

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено
Завідувач кафедри

О.В.Коваль
(ініціали, прізвище)

“ ” _____ 2019 р.

ДИПЛОМНА РОБОТА
на здобуття ступеня бакалавра

з напрямку підготовки
6.050101 “Комп’ютерні науки”

на тему: Алгоритмізації та розробки алгоритму аналізу інформації щодо визначення та прогнозування статусу гравця\команди гравців

Виконав: студент 4 курсу, групи ТР-52

Гончар Олег Вячеславович
(прізвище, ім’я, по батькові)

(підпис)

Керівник доцент, к.т.н. Коваль Олександр Васильович
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Рецензент _____
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ – 2019

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший рівень

Напрямок підготовки 6.050101 “Комп’ютерні науки”

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ О.В. Коваль
(підпис)

” ____ ” _____ 2019 р.

ЗАВДАННЯ

на дипломну роботу студенту

Гончару Олегу Вячеславовичу

(прізвище, ім’я, по батькові)

1. Тема роботи “Алгоритмізації та розробки алгоритму аналізу інформації щодо визначення та прогнозування статусу гравця\команди гравців”

керівник роботи доцент, к.т.н. Коваль Олександр Васильович

(прізвище, ім’я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від ” ____ ” _____ 201__ р.
№ _____

2. Строк подання студентом роботи _____ 201__ р.

3. Вихідні дані до роботи персональний комп’ютер під керуванням операційної системи Microsoft Windows, база даних PostgreSQL, мова програмування C#.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) проаналізувати існуючі програмні рішення та можливі засоби реалізації взаємодії, обґрунтувати обрані програмні застосунки та шляхи розробки програмних додатків, розробити програмне забезпечення, розробити користувацький інтерфейс, зробити висновки за результатами роботи

5. Перелік ілюстративного матеріалу

1. Класифікація моделей та методів прогнозування. 2. Приклади виконання прогнозування за допомогою пакету Excel. 3. Функції пакету MathCAD. 4. Інтерфейс програми RStudio. 5. Вікна додатку. 6. Модель нейронних мереж.

6. Публікації:

7. Апробації:

8. Дата видачі завдання ”__” _____ 201__ р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Вивчення та аналіз задачі		
2.	Розробка архітектури та загальної структури системи		
3.	Розробка структур окремих підсистем		
4.	Підготовка матеріалів		
5.	Програмна реалізація системи		
6.	Захист програмного продукту		
7.	Оформлення пояснювальної записки		
8.	Передзахист		
9.	Захист		

Студент

(підпис)

Гончар О.В

(прізвище та ініціали)

Керівник роботи

(підпис)

Коваль О. В.

(прізвище та ініціали)

АНОТАЦІЯ

Мета роботи — дослідження алгоритмів прогнозування часових рядів і вибір оптимальної. Для обраної задачі було обрано мову програмування C#. Для створення інтерфейсів використано засіб візуального програмування Windows Forms API. Серед досліджуваних алгоритмів було обрано лінійну регресію, модель нейронних мереж, та модель ARIMAX. Результати дослідження доповідалися на одній науковій конференції.

Записка містить 69 сторінки, 15 рисунків, 1 таблицю, 3 додатки і 26 посилань.

Ключові слова: ПРОГНОЗУВАННЯ, C#, ЧАСОВІ РЯДИ, ML.NET, NPGSQL, POSTGRESQL, ARIMAX, НЕЙРОННІ МЕРЕЖІ, ЛІНІЙНА РЕГРЕСІЯ.

ABSTRACT

The purpose of the work is to research the forecasting algorithms of time series with the further choice of the optimal one. C# programming language was chosen for selected task. Windows Forms API visual programming tools were used for creating interfaces. Among the investigated algorithms was the linear regression, the model of neural networks, and the ARIMAX model. The thesis which were written according to the materials of the work, was discussed at one scientific conference.

The note contains 69 pages, 15 figures, 1 table, 3 attachments and 26 links.
KEYWORDS: FORECASTING, C#, TIME SERIES, ML.NET, NPGSQL, POSTGRESQL, ARIMAX, NEURAL NETWORKS, LINEAR REGRESSION.

ЗМІСТ

ВСТУП.....	4
1 ПОСТАНОВКА ЗАДАЧІ АЛГОРИТМІЗАЦІЇ ТА РОЗРОБКИ АЛГОРИТМУ АНАЛІЗУ ІНФОРМАЦІЇ ЩОДО ВИЗНАЧЕННЯ ТА ПРОГНОЗУВАННЯ СТАТУСУ ГРАВЦЯ\КОМАНДИ ГРАВЦІВ.....	7
2 ПІДХОДИ ДО ВИРІШЕННЯ ПРОБЛЕМИ АНАЛІЗУ ІНФОРМАЦІЇ ЩОДО ЇЇ ПРОГНОЗУВАННЯ	9
2.1. Сімейство програмних продуктів MS Office.....	10
2.2. Пакет MatLab	10
2.3. Пакет MathCad	11
2.4. Мова програмування R	11
2.5. Мова програмування Python	11
Висновки до розділу 2	11
3 ЗАСОБИ РОЗРОБКИ	13
3.1 Середовище розробки Visual Studio 2017	13
3.2 Інтерфейс Windows Forms	15
3.3 Бібліотека ML.NET	16
3.4 Бібліотека Extreme	16
Висновки до розділу 3	16
4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ	17
4.1 Модель лінійної регресії.....	17
4.2 Модель ARIMAX	19
4.3 Модель нейронних мереж	17
4.4 Коефіцієнт детермінації.....	19
4.5 Середнє абсолютне відхилення	20
Висновки до розділу 4	20
5 РОБОТА КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ	22

5.1 Системні вимоги.....	22
5.2 Робота аналітика з програмним продуктом.....	22
Висновки до розділу 5	20
ВИСНОВКИ.....	28
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	29

ВСТУП

Проблема прогнозу являється однією із ключових проблем у подальшому розвитку спорту. Важливість розробки проблеми прогнозування у області спорту обумовлена необхідністю пошуку нових шляхів підготовки кваліфікованих спортсменів.

Прогноз є основою прийняття багатьох рішень, які надають потрібну інформацію, необхідну для подальшого розвитку спортсмена у його кар'єрі. За допомогою алгоритмів прогнозу керівники команд гравців можуть точніше виконувати свої дії над своїми гравцями [1].

В новій вітці спорту, яка називається кіберспорт, ці прогнози виявлятимуться набагато точнішими і ціннішими ніж у інших напрямках. Так усі данні щодо гравців щохвилинно оновлюються у внутрішній базі кожної з ігр, до яких належать ці кіберспортивні дисципліни. Ці данні можливо з легкістю отримати та використовувати у своїх цілях.

Головним етапом для прогнозування є виявлення найточнішого методу, який підходить до нашої області. Так ми можемо точно виділити окремі моделі прогнозування та використовувати їх у нашій системі.

Також важливим буде вибірка певної оцінки моделі, яка допоможе нам визначити, який з них виявляється найточнішим.

Вирішення питання прогнозу статусу гравців можна назвати неможливим, так як точне визначення кількості факторів та їх цінність для сприятливого прогнозу не є дуже простою задачею. Слід також оцінити людські чинники, які можуть значно відхиляти реальну поведінку гравця від прогнозованої.

Для розробки програмного продукту, який буде виконувати прогнозування статусу гравців було обрано мову програмування C#.

Проведене дослідження сприяло розробці системи порівняння та вибору алгоритмів для прогнозування даних, які базуються на часі, по гравцям, а також

дослідження ефективності кожної моделі прогнозування з певною оцінкою кожного з них.

У першому розділі описується постановка задачі алгоритмізації та розробки алгоритму аналізу інформації щодо визначення та прогнозування статусу гравця\команди гравців.

У другому розділі описуються підходи до вирішення проблеми ефективного прогнозування рейтингу гравців.

У третьому розділі вказуються основні засоби розробки системи.

У четвертому розділі дано опис реалізованого програмного продукту та його архітектури.

У п'ятому розділі описано роботу користувача з програмною системою.

1 ПОСТАНОВКА ЗАДАЧІ АЛГОРИТМІЗАЦІЇ ТА РОЗРОБКИ АЛГОРИТМУ АНАЛІЗУ ІНФОРМАЦІЇ ЩОДО ВИЗНАЧЕННЯ ТА ПРОГНОЗУВАННЯ СТАТУСУ ГРАВЦЯ\КОМАНДИ ГРАВЦІВ

Використовуючи теоретичний матеріал, наданий керівником дипломної роботи та співробітниками бази проходження практики, реалізувати інформаційну систему для ефективного прогнозування статусу гравців з кіберспортивної дисципліни CS:GO. Система повинна дозволяти користувачеві обирати список гравців, та моделі прогнозування, які будуть порівнюватись.

В результаті розробки програмного продукту, спеціалісту з спортивної аналітики буде надана можливість опрацьовувати дані різних гравців з різних команд, з усіх їх даних, отриманих з сервісу DREAMTEAM, прогнозувати їх подальшу поведінку і отримувати найбільш ефективні стратегії втручання в розвиток цього гравця\гравців. За рахунок цього спеціаліст зможе задати необхідних дій над своєю командою, вести догляд за окремими гравцями та обирати відповідну стратегії щодо покращення їх результатів.

Перед початком будь-якого аналізу треба отримати потрібну інформацію. Дані передбачатимуться у вигляді бази даних на основі об'єктно-реляційної системи керування базами даних PostgreSQL. Функції PostgreSQL дозволяють зберегти усі необхідні таблиці з даними на сервері, та отримувати їх для подальшої обробки клієнтом.

Наступною складністю задачі є вибір мови програмування, на якій буде реалізоване зчитування даних та реалізація аналізу інформації щодо її прогнозування. Так вибір пав на мову програмування C#. Ця мова є сучасною об'єктно орієнтованою мовою, яка завжди еволюціонує та підтримується. Задача

пошуку потрібних бібліотек та впровадження їх обраною мовою не складатиме непотрібної важкості.

Система повинна мати зручний інтерфейс для перегляду інформації по кожному з гравців, та надавати можливість переглядати одразу декількох з них у вигляді команди, яка може бути як заздалегідь сформованою, так і несформованою. Залежно від складності проектування інтерфейсу існують такі технології його реалізації мовою C#:

- система Windows Presentation Foundation — графічна підсистема, яка починаючи з .NET Framework 3.0 в складі цієї платформи;
- інтерфейс Windows Forms — інтерфейс програмування додатків , відповідальний за графічний інтерфейс користувача і є частиною Microsoft .NET Framework;
- проект Windows Console Application. Даний тип проекту зовсім не підходить для розроблювальної системи через складність та неможливість повного відображення інформації користувачеві.

Серед перерахованих технологій було обрану технологію Windows Forms. Головною причиною даного вибору є можливість створення зручного інтерфейсу програми не виконуючи складних операцій для його створення.

Головним кроком розробки системи є створення вибірки алгоритмів прогнозування. Для цього слід виконати класифікацію існуючих методів. В основному методи прогнозування поділяються на дві категорії:

- інтуїтивні методи (ті, які мають справу з судженням);
- формалізовані методи (ті, які мають справу зі деякими математичними моделями)

Зазначимо, що в системі будуть використовуватися лише ті данні, які були отримані із сервісу Dreamteam. Тому інтуїтивні методи не розглядатимуться в рамках цієї задачі.

Формалізовані ж методи в свою чергу поділяються на моделі предметної області, тобто ті, що використовуються в термодинаміці, механіці, тощо, та моделі

часових рядів, які шукають залежність в деякому процесі. З цих двох моделей ми будемо використовувати саме модель часових рядів [8].

Класифікація моделей часових рядів:

- статистичні моделі
- структурні моделі

У статистичних моделях залежність майбутнього значення від минулого задається в вигляді деякого рівняння. До них відносяться:

- регресивні моделі (лінійна регресія, нелінійна регресія);
- авторегресійні моделі (ARIMAX, GARCH, ARDLN);
- модель експоненціального згладжування;
- модель за вибіркою максимального подібності [25];

У структурних моделях залежність майбутнього значення від минулого задається у вигляді певної структури і правил переходу по ній [24]. До них відносяться:

- нейромережеві моделі;
- моделі на базі ланцюгів Маркова;
- моделі на базі класифікаційно-регресійних дерев;

Переглянувши детально цю класифікацію ми можемо одержати такий її загальний вигляд (рисунок 1.1):

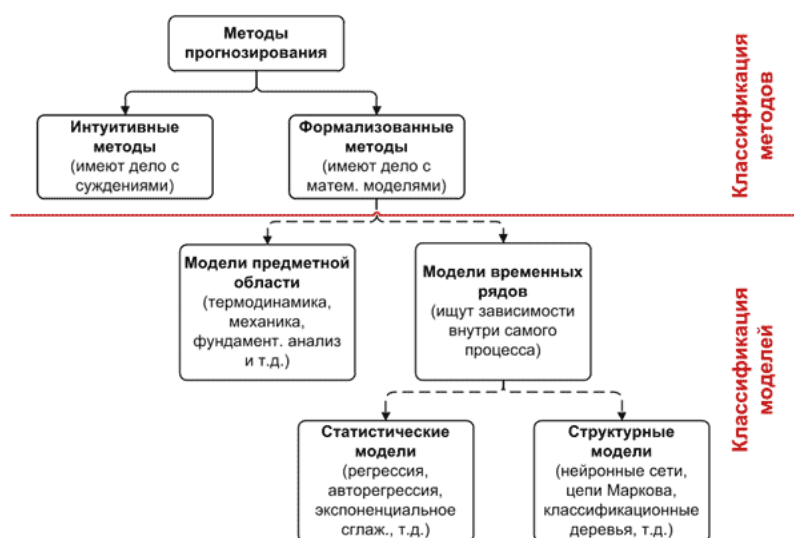


Рисунок 1.1 — Класифікація моделей та методів прогнозування.

Саме для обраної задачі слід відзначити такі моделі прогнозування [26]:

- лінійна регресивність;
- модель нейронних мереж;
- модель ARIMAX;

Для перевірки ефективності зазначених моделей будемо шукати відхилення відповідно до кожної з них. В системі в якості оцінки відхилення прогнозованих даних від реальних будемо використовувати такі метрики:

- коефіцієнт детермінації (позначається як R^2 — R-квадрат);
- середнє абсолютне відхилення;

Розроблена система повинна забезпечувати наступні можливості:

- зчитування вхідних даних з бази даних Postgre;
- робити вибірку окремих гравців;
- переглядати всю необхідну інформацію щодо цих гравців;
- відображення усієї інформації на графіку з калібрування окремих частин форми;
- вибір потрібних моделей прогнозування;
- виконання прогнозу, відповідно до обраних користувачем моделей та деякому обраному списку гравців;
- ефективне відображення даних прогнозу на загальному графіку даних гравців, регулювання моделей прогнозування та відповідне відображення різними кольорами різних моделей прогнозування, можливість виконувати дії приближення \ віддалення на вікні графіку;
- вивід та регулювання кожної з оцінок алгоритмів прогнозування, виділення окремих місць їх розташування;
- вивід даних по прогнозування на кожний день після останнього дня реальних даних;
- порівняння ефективності методів прогнозування;

Метою розробки інформаційної системи аналізу інформації щодо її прогнозування є створення програмного продукту, що дасть змогу отримати дані, спираючись на які, спеціалісти в області кіберспорту зможуть більш ефективно

виконувати менеджмент гравців. Така система надає можливість оцінювати кожного гравця у майбутньому. Також ця система надаватиме нові можливості для людей, які зацікавлені у розвитку цієї гілки спорту.

Мета цього проекту полягає в тому, щоб дати змогу аналітику у спортивній сфері з мінімальним досвідом користування комп'ютером здійснити аналіз вибірки гравців, яка знаходиться у базі даних глобальної системи перегляду за гравцями, щодо прогнозування статусу гравця, або команди гравців, при цьому потребуючи жодної інформації про алгоритми прогнозування. Цей проект дає змогу аналітикам дізнатися найефективнішу стратегію пошуку рішень щодо відведених до них гравців, або команд. А також порівняти обрані моделі прогнозування відповідно до окремих випадків гравців.

2 ПІДХОДИ ДО ВИРІШЕННЯ ПРОБЛЕМИ АНАЛІЗУ ІНФОРМАЦІЇ ЩОДО ЇЇ ПРОГНОЗУВАННЯ

На даний момент у вільному доступі, або за допомогою окремих веб—сервісів, існує можливість отримати усі дані про гравців прямо з ігр. Вони досить широкі та містять велику кількість інформації по усім гравцям. Серед цих даних також є відомості про рейтинг кожної людини на окремий момент часу. Ці бази даних формують і зберігають окремі веб—сервіси. Потрібну інформацію можливо отримати відповідно від володарів цих сервісів, які займаються аналізом інформації по гравцям.

В зв'язку зі складною предметною областю, рішень, які б повністю відповідали завданню, приведеному вище, просто не існує. Такі системи не впроваджуються через значну кількість факторів впливу на окремого гравця, що ці данні загалом виявляються неточними. Ще однією причиною є дуже великі дані по усій базі гравці, які збільшують час для її обробки [9].

Більшість систем для аналізу інформації в області кіберспорту використовуються для швидкого доступу і перегляду інформації по окремим матчам, не спираючись на великі бази даних гравців, а лише на дані, які можна отримати одразу з гри. Такі системи не надають можливості перегляду прогнозу на майбутнє. Також інша частина цих систем використовуються для створення соціальних мереж для геймерів, з можливістю перегляду усієї необхідної статистики, створення команд, комунікації, тощо. Серед таких систем також прогнозування не набирає багато популярності.

Однак серед всіх прикладів вже розроблених систем можна відокремити одну велику гілку, яка використовує прогнозування в своїх сервісах для збагачення компанії на ставках. Такі прогнози виконується в загалом за допомогою окремих

програм. Для кожної з програм аналітик вибирає потрібні дані для аналізу та виконує прогноз, з подальшою можливістю його корегування.

Такий вид прогнозування виконується над іншою областю даних та його не можна віднести до поставленої задачі, так як загалом ці прогнози виконуються над майбутніми матчами і спираються на попередню історичну інформацію усіх завершених матчів команд. Такі системи також не можна поставити як приклад виконання обраної задачі аналізу інформації щодо її прогнозування по статусу гравців.

Загалом для аналізу та обробки інформації в цих сервісах використовуються пакети MS Office, MatLab або MathCad. Їх вистачає для невеликої кількості підрахунків та представлення їх у доступному вигляді [23]. Ці програмні пакети не являються ключовими для розв'язку нашої задачі через їх незручність для користувачів.

Якщо не привертати всю увагу на вже розроблені великими компаніями пакети, які так популярні серед аналітиків, то слід розглянути і інші засоби аналізу інформації. Так для визначення та прогнозування даних по гравцям можна застосовувати мову програмування R.

Програмних продуктів, спроможних спрогнозувати статус гравців в наш час у вільному доступі взагалі немає. А це лишається основною задачею перед інформаційною системою, що була описана вище.

Аналіз інформації по визначенню статусу гравців та прогнозування їх подальшого розвитку в запропонованому програмному продукті спираються на принципи індивідуально-орієнтовного моделювання.

2.1 Сімейство програмних продуктів MS Office

Існує велика кількість програмного забезпечення, що спеціалізується саме на цьому напрямку. На жаль, далеко не всі користувачі знають, що звичайний табличний процесор Excel має у своєму арсеналі інструменти для виконання

прогнозування, які за своєю ефективністю мало чим поступаються професійними програмами.

Пакет Microsoft Office був створений корпорацією Microsoft та доступний для платформ Microsoft Windows та Mac OS. Перше видання пакету Microsoft Office відбулося у 1990 році. З того часу програмний пакет майже щорічно оновлюється і отримує новий функціонал [10].

За допомогою програми для роботи з електронними таблицями Microsoft Office Excel ми маємо можливість провести статистичний аналіз даних. Пакет має відповідні функції для перегляду даних та надає можливість виконувати над ними аналіз. Зокрема у пакеті є можливість використання моделі регресивного моделювання, що має відповідну важливість для нашої задачі [10].

Слід також згадати, що окрім офіційних розробників компанії Microsoft над цим пакетом також виконують роботу різні спеціалісти з усіх кінців світу. Так для облегшення та збільшення функціоналу пакетів ми маємо можливість встановлювати так звані Add-Ins, додаткові плагіни для роботи з Microsoft Excel. В цих плагінах ми можемо знайти багато алгоритмів прогнозування.

Стандартні функції Microsoft Office Excel включають в себе такі види апроксимації (рисунок 2.1):

- лінійна;
- логарифмічна;
- експоненціальна;
- степенева;
- поліноміальна;
- лінійна фільтрація;

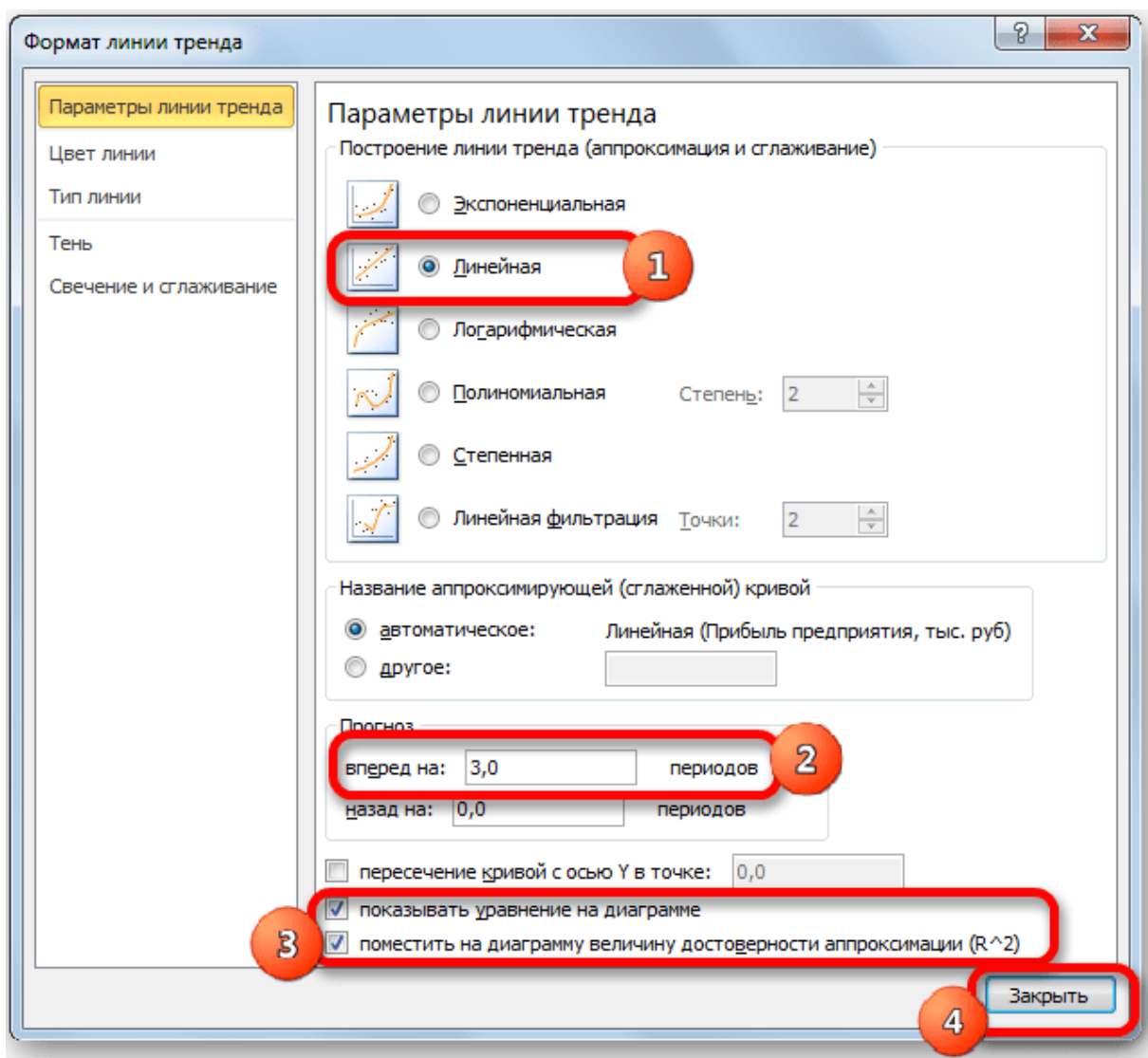


Рисунок 2.1 — Приклад прогнозування для відтворення даних на графіку за допомогою лінії трендів в пакеті Microsoft Excel.

Екстраполяцію для табличних даних можна зробити через стандартну функцію Excel ПРЕДСКАЗ. Цей аргумент відноситься до категорії статистичних інструментів і має наступний синтаксис (рисунок 2.2):

=ПРЕДСКАЗ(Х;реальні_значення_у;реальні_значення_х)

Також в пакеті існують такі функції вирішення задачі для прогнозування:

- оператор ТЕНДЕНЦИЯ;
- оператор РОСТ;
- оператор ЛИНЕЙН;
- оператор ЛГРФПРИБЛ;

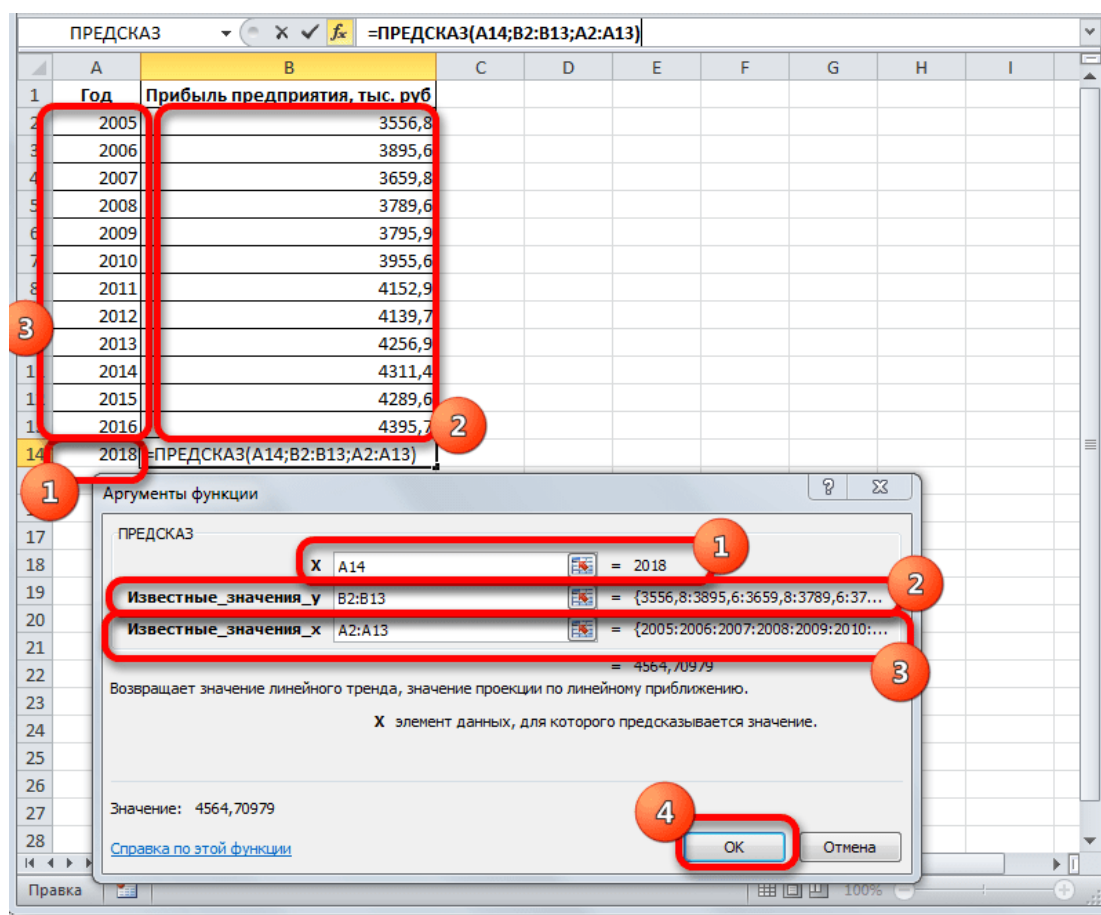


Рисунок 2.2 — Приклад прогнозування та відтворення даних за допомогою функції ПРЕДСКАЗ в пакеті Microsoft Excel.

Пакет надає можливість підстроювати прогноз під користувача та загальна кількість параметрів може спочатку його збентежити.

Та функціоналу Excel не вистачає для відтворення поставлених вище задач. Загалом все сімейство програм Microsoft Office призначено для розв'язання великого спектру завдань. Але вони зовсім не допомагають при розробці складних систем. Тому Microsoft Office Excel не допоможе вирішити проблему аналізу інформації щодо визначенню та прогнозуванню статусу гравців.

Ціна останньої версії базового пакету програм складає приблизно 120 доларів США. Це є ще одним фактором, через який сімейство програм Microsoft Office важко розглядати як систему для вирішення задачі описаної вище, адже впровадження цього програмного забезпечення потребує відносно значного фінансування.

2.2 Пакет MatLab

Пакет MatLab являє собою пакет прикладних програм для числового аналізу, а також мовою програмування, що також використовується в цьому пакеті. Система створена компанією The MathWorks і є зручним засобом роботи з математичними матрицями, малювання функцій, роботи з алгоритмами, створення робочих оболонок з програмами в інших мовах програмування.

Фінансові інженери, аналітики та економісти по всьому світу використовують можливості інструментів компанії MathWorks для аналізу часових рядів при прогнозуванні волатильності ринку, аналізу кореляції в серії даних, перевірки гіпотези про динаміку ринку та побудови моделей для подальшого аналізу або моделювання майбутніх результатів [2].

Мова програмування пакету MatLab дозволяє налаштовувати програму не гірше, ніж будь-які інші мови програмування. Спеціальні засоби, які працюють з програмним забезпеченням Maple роблять його повноцінною системою для роботи з алгеброю .

Інструменти MatLab призначені для фінансової сфери дозволяють отримати доступ до даних, візуалізувати і аналізувати історичні та поточні дані часових рядів, для визначення закономірності та виконання прогнозування [3].

Середовище для аналізу MATLAB включає засоби для збору даних і автоматизації завдань. За допомогою MATLABiSimulink можливе управління даними і можливість їх отримання з підключених плат збору даних, вимірювальних інструментів, веб-камер і плат захоплення зображень. Також є можливість відправляти і отримувати повідомлення через CAN-шини.

Одержані дані можна досліджувати, виробляти їх візуалізацію в реальному часі і проводити їх аналіз. Після цього з'являється можливість створювати тестові системи, перевіряти розробки і концепти, автоматизувати повторювані задачі. Створені в MATLAB тестові додатки запускаються на інших комп'ютерах.

Інструменти MATLAB, призначені для фінансової сфери, дозволяють отримувати доступ, візуалізувати і аналізувати історичні та поточні дані часових

рядів, щоб виявити закономірності або розкрити складні відносини між ними. У єдиному обчислювальному середовищі є можливість [11]:

- отримати доступ до даних з різних джерел, включаючи файли, електронні таблиці, бази даних, постачальників даних в інтернеті;
- зберігати даних в об'єктах фінансових часових рядів, щоб спростити управління даними, перетворення даних і обробки відсутніх даних;
- виконувати технічний аналіз з різними фільтрами, осциляторами, стохастикою і індексами;
- створювати призначені для користувача процедури аналізу, візуалізації і анімації для захоплення, уявлення і передачі процесу аналізу і відкриття;
- налаштування середовища аналізу з розширеною функціональністю для обробки сигналів, статистики або економетрики;

За допомогою пакету MatLab можливо побудувати і оцінити параметри будь-якої моделі, використовуючи традиційні методи, такі як рішення систем звичайних диференціальних рівнянь, виконання багатовимірної регресії, або з використанням методів оптимізації для підбору моделей за даними часових рядів. Крім того, ви можете звернутися до спеціалізованих методів моделювання, таким як ARMAX / GARCH, VAR / VARMA і лінійні або нелінійні стохастичні диференціальні рівняння [22].

MATLAB дозволяє змішувати і поєднувати різні методи і підходи до розробки моделей. Ви можете комбінувати безперервним і дискретним часом, наближаючись до випадкових / дискретних подій. Ці типи моделей використовуються для імітації торгових систем, які враховують обмеження біржових торгів або описують макроекономічні системи, що містять випадкові події.

Але сам інструмент надає можливість робити це все для спеціалістів, які розбираються саме в цьому інструменті. Складність використання цього пакету з іншими мовами програмування також являє собою великий мінус. Тому програмне забезпечення сімейства програмних продуктів MatLab не дає змоги для повного розв'язання потрібних завдань нашої розроблювальної системи.

2.3 Пакет MathCad

Система комп'ютерної алгебри MathCad, орієнтована на підготовку інтерактивних документів з обчисленнями і візуальним супроводженням. MathCad був задуманий і створений Алленом Раздовим з Массачусетського технологічного інституту. Перша версія програми була випущена у 1984 році. Після цього вона постійно оновлюється.

MathCad підтримує зчитування даних з багатьох інших пакетів роботи з даними, такими як Microsoft Excel та він не здатен зчитувати дані з бази даних Postgre. Існують шляхи обходу цієї проблеми, такі як створення засобів перетворення даних з бази даних у файли формату Microsoft Excel та роботу з ними. Але це є дуже моторним процесом і не його не можна назвати найефективнішим способом.

MathCad надає можливість працювати зі статистичними даними. Для аналізу статистичних даних можливо виконання його графічного представлення, виведення окремих значень. В ньому представлений широкий спектр функцій для роботи з різними статистичними вибірками, зокрема, реалізовано регресійний і кореляційний аналіз [4].

При обробці експериментальних даних в Mathcad з метою дослідження їх природи, виникає необхідність висловити залежну змінну у вигляді деякої математичної функції від однієї або декількох незалежних змінних. Дана залежність отримала назву регресійна модель або рівняння регресії, а методи, що дозволяють отримати цю залежність, прийнято називати методами регресійного аналізу в Mathcad.

Методи регресійного аналізу дозволяють: проводити розрахунок різного виду регресійних моделей; перевіряти гіпотезу адекватності моделі наявними спостереженнями; використовувати модель для прогнозування значень залежної змінної при нових значеннях незалежної змінної.

У Mathcad існує набір функцій, що дозволяє розрахувати різні регресійні моделі. На рисунку представлені функції, що використовуються при створенні регресійних моделей (рисунк 2.3):

Наименование модели	Вид уравнения регрессии	Функция Mathcad
Линейная	$y(x) = a \cdot x + b$	<i>line(vx, vy)</i>
Полиномиальная n-ой степени	$y(x) = \sum_{i=0}^n a_i \cdot x^i$	<i>p1:=regress(vx,vy,n)</i> <i>interp(p1,vx,vy,x)</i>
Фрагменты полиномов 2-ой степени	$y_i(x) = \sum a_i \cdot x^2 + b_i \cdot x + c_i$	<i>p2:=loess(vx,vy,span)</i> <i>interp(p2,vx,vy,x)</i>
Экспоненциальная	$y(x) = a \cdot e^{b \cdot x} + c$	<i>expfit(vx,vy,g)</i>
Логистическая функция	$y(x) = a / (1 + b \cdot e^{c \cdot x})$	<i>lgsfit(vx,vy,g)</i>
Синусоидальная	$y(x) = a \cdot \sin(x + b) + c$	<i>sinfit(vx,vy,g)</i>
Степенная	$y(x) = a \cdot x^b + c$	<i>pwfit(vx,vy,g)</i>
Логарифмическая	$y(x) = a \cdot \ln(x + b) + c$	<i>logfit(vx,vy,g)</i>
Логарифмическая короткая	$y(x) = a \cdot \ln(x) + b$	<i>lnfit(vx,vy,g)</i>

Рисунок 2.3 — Функції пакету MathCAD для роботи з регресивними моделями.

Після визначення регресійних залежностей в Mathcad, актуальним є вибір з їх сукупності найкращою функції, з точки зору адекватності опису вихідних експериментальних даних. Як критерій, що дозволяє вибрати найкращу регресійну модель, пропонується використовувати коефіцієнт детермінації, чисельно рівний коефіцієнту кореляції в квадраті. Значення коефіцієнта кореляції в Mathcad дозволяє розрахувати функція *corr* (A, B), де A і B - два вектора значень.

Моделі прогнозування, отримані при використанні різних модифікацій методу Хольта, методу Брауна, функцій *predict* і *loess* в MathCAD, а також за допомогою методу ковзних середніх в Excel, мають приблизно однакову здатність прогнозування.

Та, згідно нашої задачі, функціонал MathCad також не дозволяє створити на його базі систему, яка здатна аналізувати та прогнозувати статус гравців. Це програмне рішення підтримує формати MathCad Document, файли формату XML та файли бінарного формату для читання та запису. Також сам інтерфейс може бути незрозумілим для недосвідченого комп'ютерного користувача. А ціна в 700 доларів

США на MathCad в базовій комплектації є надто високою для підприємств, які займаються аналітикою в медичній галузі.

2.4 Мова програмування R

Мова R — програмне забезпечення та мова програмування, що дозволяє ефективно вивчати часові ряди. Для роботи з цією мовою існує спеціальне програмне забезпечення RStudio, яке дозволяє користувачеві створювати код для реалізації деякого аналізу інформації. Це забезпечення дозволяє переглядати процес виконання коду в консолі, та спостерігати за даними в окремому вікні. Також в цьому програмному забезпеченні є вбудований сервіс для доступу до вікі—сторінок, для швидкого знаходження потрібних команд цієї мови (рисунк 2.4).

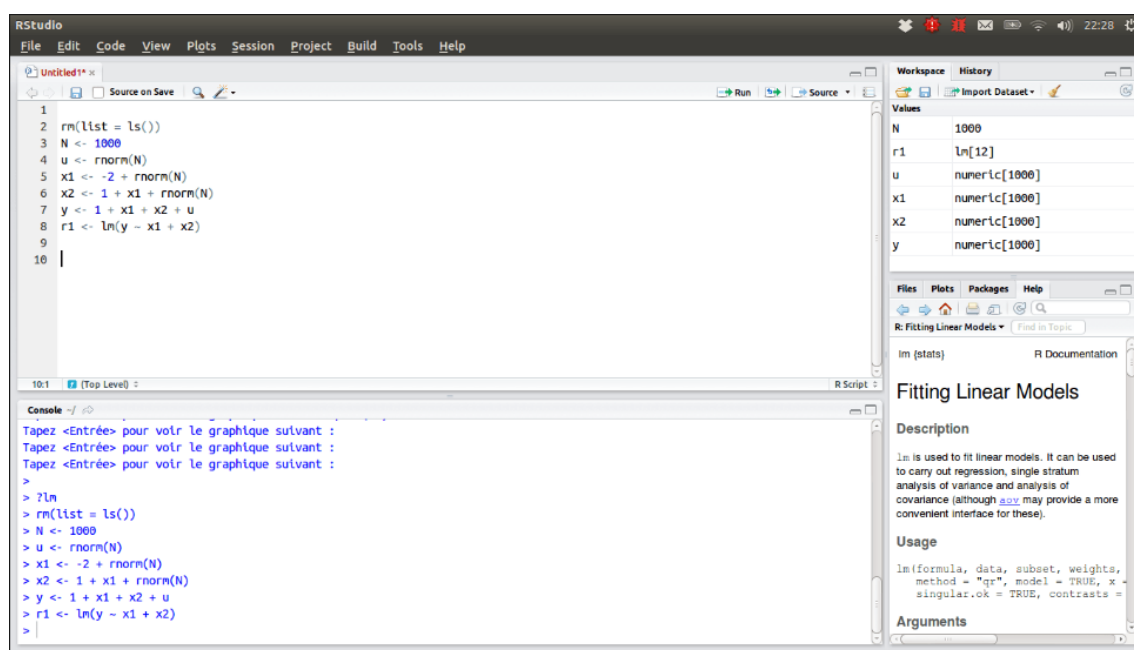


Рисунок 2.4 — Інтерфейс програми RStudio.

Автоматичний прогноз часових рядів (одновимірний чи ні) дуже корисний для поставленої задачі. Часто досить важко відстежити відповідну підготовку для створення моделей часових рядів, тому автоматичний алгоритм прогнозування є важливим інструментом. Автоматичний алгоритм прогнозування повинен визначити

відповідну модель часових рядів, оцінити параметри моделі і обчислити прогноз [12].

Найбільш використовувані автоматичні алгоритми прогнозу засновані на експоненціальному згладжуванні або моделях ARIMA.

Всі методи є дієвими в певних обставинах, але якість прогнозу пов'язана з вибором моделі. Розрахунки потребують часу і часто є сумнівними для придатності моделі. Взагалі, немає жодного методу, який краще виконує всі часові ряди. Для часових рядів з різними специфікаціями існують різні методи, які виконуються більш ефективно. Для використання методів прогнозування на мові R потребує встановлення деякого пакету статистичних даних [12].

В багатьох випадках дані для доступу мовою R реалізуються в форматі Excel. Робота з цими даними в середовищі R означає, що вони повинні бути збережені у форматі csv. При роботі з часовими рядами в R, спочатку дані повинні бути перетворені в формат часових рядів, так щоб мова R могла їх розпізнавати.

Алгоритми та фреймворки моделювання для автоматичного одновимірного прогнозування часових рядів реалізовані в пакеті forecast в R. Функція forecast є загальною і має методи S3 для широкого діапазону моделей часових рядів. Ця функція обчислює точні прогнози та інтервали прогнозування з моделі часових рядів. Методи існують для моделей, встановлених за допомогою ets, auto.arima, Arima, arima, ar, HoltWinters і StructTS [13].

Також існує метод для об'єкта ts. Якщо об'єкт часових рядів передається як перший аргумент forecast, функція буде виробляти прогнози на основі експоненціального алгоритму згладжування.

- Вихід з функції forecast є об'єктом класу «прогноз» і включає в себе
- найменше наведеної нижче інформації:
- оригінальний ряд;
- точкові прогнози;
- інтервали прогнозування зазначеного покриття;
- використовуваний метод прогнозування та інформація про відповідну модель;

- залишки з встановленої моделі;
- однокрокові прогнози з пристосованої моделі за період спостережуваних даних

Мова R являє собою дуже потужний інструмент для роботи з даними. Багато бібліотек реалізовано для цієї мови програмування, однак для користувача системи, яка розробляється, така мова зовсім не підходить через складний інтерфейс та потребою знати всі потрібні команди для виконання задачі аналізу. До того ж назвати код, написаний мовою R, окремою системою не є правильним. Ця мова підійде для локальних рішень компаній, які займаються ринком чи іншою складною інформацією.

2.5 Мова програмування Python

Мова Python є інтерпретованою, об'єктно-орієнтованою мовою програмування високого рівня з динамічною семантикою. Її вбудовані в високий рівень структури даних, в поєднанні з динамічною типізацією і динамічним зв'язуванням, роблять його дуже привабливим для швидкого розробки додатків, а також для використання в якості сценарію або мови клею для з'єднання існуючих компонентів.

Простий, легкий у вивченні синтаксис Python підкреслює читабельність і, отже, знижує вартість технічного обслуговування. Python підтримує модулі та пакети, що заохочує модульність програми та повторне використання коду. Інтерпретатор Python і велика стандартна бібліотека доступні у вихідній або двійковій формі без нарахування всіх основних платформ і можуть вільно розповсюджуватися.

Екосистема Python зростає і може стати домінуючою платформою для прикладного машинного навчання.

Основна причина для прийняття Python для прогнозування часових рядів полягає в тому, що це мова програмування загального призначення, яку можна використовувати як для R&D, так і для виробництва [14].

Мова Python широко використовується для машинного навчання і наука про дані через відмінну бібліотечну підтримку. Вона швидко стала однією з домінуючих платформ для фахівців у галузі машинного навчання та наукових даних і користується більшим попитом, ніж навіть платформа R роботодавцями (рисунок 2.5).

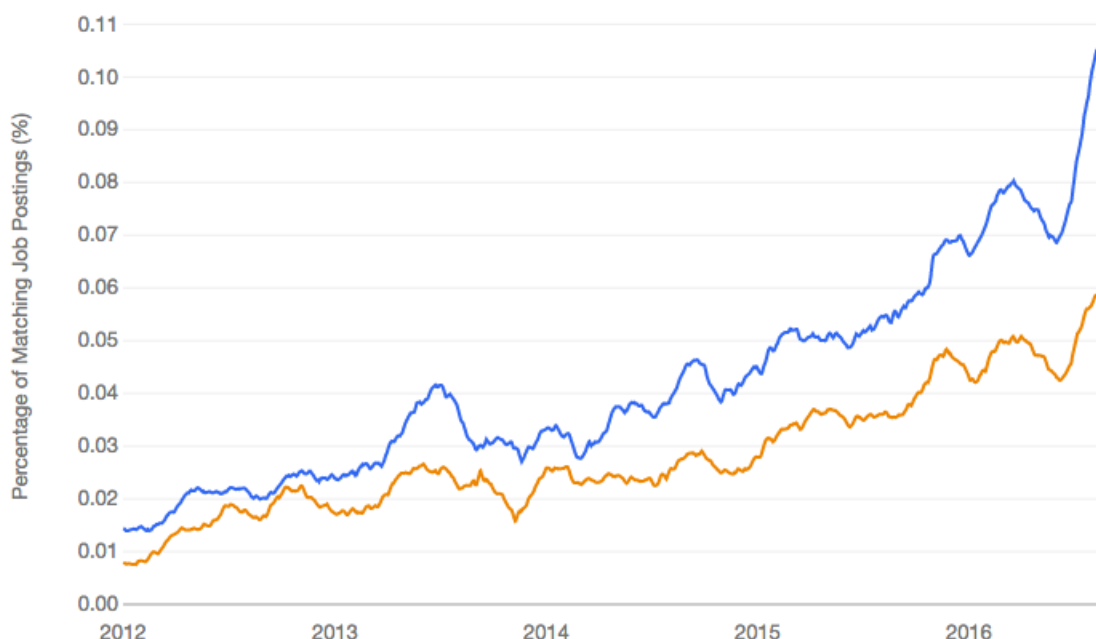


Рисунок 2.5 — Графік співвідношення робіт з машинним навчанням на Python та робіт з машинним навчанням на R.

Додаток SciPy - це екосистема бібліотек Python для математики, науки і техніки. Це доповнення до Python, яке вам потрібно для прогнозування часових рядів.

Дві бібліотеки SciPy є основою для більшості інших; вони є NumPy для забезпечення ефективних операцій масиву та Matplotlib для побудови даних. Існують також три бібліотеки SciPy вищого рівня, які забезпечують ключові можливості для прогнозування часових рядів у Python [20].

Ця мова добре підходить у разі розробки серверної частини системи, де основні функції будуть на ній описані, та викликатись клієнтом. Ця мова програмування не була обрана через складність її інтегрування з іншими мовами.

Висновки до розділу 2

У даному розділі були описані існуючі технології для побудови обраної системи аналізу інформації щодо її прогнозування. Звичайно таких технологій може бути набагато більше. Для кожної технології були описані причини та наслідки використання, разом з детальним описом функцій, які допомогли б в розробці даної системи.

3 ЗАСОБИ РОЗРОБКИ

Важливим етапом розробки інформаційної системи є вибір засобів для програмної реалізації та підключення необхідних технологій до проекту. Для написання коду програми було обрано інтегровану середу розробки Microsoft Visual Studio Community 2017. Для створення графічного інтерфейсу системи використовувався інтерфейс програмування додатків Windows Forms, що є частиною Microsoft .NET Framework.

Задля ефективної реалізації алгоритмів використовувалась об'єктно-орієнтована мова програмування C#.

Для зчитування вхідної інформації в систему та створення з бази даних була використана зовнішня бібліотека Npgsql, а для побудови графіків та їх відображення у системі — System.Drawing.

3.1 Середовище розробки Visual Studio 2017

Середовище розробки Microsoft Visual Studio — лінійка продуктів компанії Microsoft, що включають інтегроване середовище розробки програмного забезпечення та ряд інших інструментальних засобів. IDE MS Visual Studio .NET — це набір інструментів і засобів розробки різного роду застосувань (консольних, Windows, мобільних, Web-застосувань) та сервісів. MS Visual Studio є мультипрограмним середовищем, що підтримує декілька мов програмування, зокрема, C++, C#.

Довідкові відомості про всі розробки компанії Microsoft зібрані в один програмний продукт MSDN.

У MS Visual Studio кожне окреме застосування є рішенням, що складається з одного чи декількох проектів. Одночасно можна відкрити тільки одне рішення, при

роботі над кількома рішеннями одночасно потрібно запускати декілька вікон Visual Studio.

Засіб Visual Studio надає шаблони для проектів найбільш поширених типів. Використання проектів і їх шаблонів дозволяє користувачеві зосередитися на реалізації окремої функції, в той час як проект буде виконувати загальне управління та завдання побудови.

Для створення нового проекту використовується майстер застосувань. Майстер створення програм надає користувальницький інтерфейс для створення проекту за шаблоном та створення шаблонів для файлів вихідних текстів. Майстер налаштовує структуру програми, основні меню і панелі інструментів, забезпечує включення деяких заголовних файлів. З кожним роком інструментарій Visual Studio оновлюється і постійно розширюється. А компанія Microsoft постійно намагається покращити зручність середовища розробки, з урахуванням побажань розробників з усього світу, які користуються Visual Studio. В результаті понад 30 мільйонів завантажень на сьогоднішній день, що робить цей продукт одним із найпопулярніших серед розробників додатків.

Для розробників існують спеціальні безкоштовні версії Microsoft Visual Studio, які дозволяють використовувати весь потенціал цієї серії розробки.

Елементи графічного інтерфейсу MS Visual Studio характерні для Windows-програм. Розмір і форма вікон визначається конкретною конфігурацією системи. Користувач має можливість змінювати розмір і розташування окремих елементів, згортати їх з тим, щоб збільшити місце для інших, необхідних в даний момент, елементів. Це дозволяє пришвидшити розробку додатків для кожного окремого розробника, який має свої примхи щодо найефективнішого налаштування свого робочого місця.

Серед нововведень MS Visual Studio можна назвати багато сервісів, один з яких є MS Azure. Це хмарна платформа та інфраструктура корпорації Microsoft, призначена для розробників застосунків хмарних обчислень і покликана спростити процес створення онлайн-додатків.

Це середовище застосовується для розробки додатків на більшість з існуючих нині платформ.

3.2 Інтерфейс Windows Forms

Інтерфейс Windows Forms — інтерфейс програмування додатків, який відповідає за графічний інтерфейс користувача і є частиною Microsoft .NET Framework. Даний інтерфейс спрощує доступ до елементів інтерфейсу Microsoft Windows за допомогою створення обгортки для Win32 API в керованому коді. Хоча Windows Forms є лише обгорткою Windows API-компонентів ця технологія надає можливість розробки кросплатформенного графічного користувацького інтерфейсу. Його класи можна використовувати при розробці на інших мовах, таких як C++.

В наш час Windows Forms використовується як заміник старої і складнішої бібліотеки MFC, що була написана на мові програмування C++. Але розглядаючи Windows Forms варто зазначити, що вона не пропонує парадигму, як це робила MFC.

Технологія Windows Forms — це інтелектуальна технологія клієнта для .NET Framework, набір керованих бібліотек, що спрощують звичайні завдання додатків, такі як читання і запис у файловій системі. Під час використання середовища розробки, наприклад, Visual Studio, можна створити програми смарт-клієнта Windows Forms, які відображають інформацію, запитують вхідні дані користувачів і зв'язуються з віддаленими комп'ютерами через мережу.

У Windows Forms форма — це візуальна поверхня, на якій відображається інформація для користувача. Зазвичай ви створюєте додатки Windows Forms, додаючи елементи керування до форм і розробляючи відповіді на дії користувача, такі як клацання миші або натискання клавіш. Управління — це дискретний елемент інтерфейсу користувача, який відображає дані або приймає дані.

Коли користувач робить щось із вашою формою або одним з елементів керування, дія генерує подію. Ваша програма реагує на ці події за допомогою коду та обробляє події, коли вони відбуваються.

3.3 Бібліотека ML.NET

Бібліотека ML.NET — це безкоштовна бібліотека для навчання програмного забезпечення для мов програмування C #, F # і VB.NET. Також вона підтримує моделі Python при використанні разом з NimbusML. Попередній випуск ML.NET включав перетворення для інженерних розробок, таких як створення n-грамів, і учнів для обробки двійкової класифікації, багатокласової класифікації та завдань регресії. З того часу були додані додаткові завдання, такі як виявлення аномалій і системи рекомендацій [5].

Ця бібліотека надає змогу збудувати модель нейронних мереж попередньо відповідно до запропонованих даних з бази даних, або з файлів формату .csv . Та для нашого випадку ми будемо використовувати класи цієї бібліотеки для формування моделі всередині нашої системи

Основним плюсом для обраної бібліотеки є те, що вона орієнтована на об'єктно орієнтовану мову програмування C#. Так першим ж кроком для її використання ми повинні створити класи відповідних розглянутих даних.

Головним класом з яким ми будемо працювати це клас MLContext. В ньому зосереджені усі необхідні операції ML.NET. Після створення об'єкту цього класу, ми маємо засоби для створення компонентів для підготовки даних, інженерної функцій, навчання, прогнозування, оцінки моделі. Вона також дозволяє вести журнал, керувати виконанням і можливість встановлювати повторювані випадкові числа.

Інтерфейс ITransformer та похідні від нього інтерфейси дозволяють трансформувати дані у потрібний для бібліотеки вигляд.

Простір імен Microsoft.ML.Trainers містить у собі тренувальники, параметри моделей та деякі інші утиліти.

3.4 Бібліотека Extreme

Чисельні бібліотеки Extreme Optimization для .NET є міцною основою для ваших чисельних обчислювальних потреб на платформі .NET. Вона реалізує широкий набір алгоритмів, що охоплюють широкий спектр чисельних методів, включаючи: лінійну алгебру, комплексні числа, чисельну інтеграцію та диференціювання, розв'язування рівнянь, оптимізацію, випадкові числа, регресію, ANOVA, статистичні розподіли, перевірки гіпотез.

Класи в The Extreme Optimization Numerical Libraries для .NET і відносини між ними відповідають нашим повсякденним поняттям. Ця бібліотека впроваджує найкращі алгоритми, доступні сьогодні, щоб надати надійний, швидкий набір інструментів.

За допомогою Extreme Optimization Numerical Libraries для .NET можна скоротити час розробки та зосередитися на проблемі реалізації зручної комунікації з користувачем, не зволікаючись на проблеми розробки існуючих алгоритмів.

Бібліотека Extreme в нашій системі містить функції та методи для роботи з моделлю ARIMAX.

Простір імен Extreme.Statistics.TimeSeriesAnalysis містить класи для роботи з даними часових рядів.

Клас ArimaModel має два конструктора. Перший приймає три аргументи і використовується для побудови моделі ARMA (p, q). Перший аргумент — це вектор, який містить дані часових рядів. Другий і третій аргументи — порядок авторегресії і ковзної середньої величини відповідно. Наступний створює ARMA (2,1), або еквівалентно, модель ARIMA (2,0,1) для вектора у часових рядах.

Метод Compute оцінює параметри моделі шляхом мінімізації точної функції максимального правдоподібності.

Параметри моделі можна отримати через збірку параметрів. Першими параметрами p є коефіцієнти авторегресивних компонентів. Наступними q параметрами є коефіцієнти компонентів ковзних середніх. Якщо середнє значення також було оцінено, воно повертається як останній параметр. Коефіцієнти

авторегресії і ковзних середніх також можуть бути отримані окремо через властивості `AutoRegressiveParameters` і `MovingAverageParameters`.

Члени цих колекцій мають тип `Parameter`, і можуть бути використані для отримання широкого діапазону інформації про обчислених значеннях, включаючи стандартну помилку, значимість тестів і довірчі інтервали.

Після обчислення моделі для прогнозування нових значень можна використовувати метод `Forecast`. Цей метод має три перевантаження.

Без аргументів метод повертає прогноз на один крок вперед на основі обчисленої моделі. За допомогою одного аргументу він обчислює прогноз точки для заданої кількості кроків вперед. Вона повертає `VectorT`, що містить прогноз точок на вказану кількість періодів.

Висновки до розділу 3

У даному розділі було описані засоби розробки системи. Були наведені середовище розробки `Visual Studio`, засоби мови програмування `C#`, та бібліотеки, які використовувались у проєкті.

4 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

В даному розділі міститься перелік використаних обчислювальних методів. А також опис використаних обчислювальних методів і математичні та програмні засоби їх реалізації.

4.1 Модель лінійної регресії

У статистиці лінійна регресія — це метод моделювання залежності між скаляром y та векторною (у загальному випадку) змінною X . У разі, якщо змінна X також є скаляром, регресію називають простою.

При використанні лінійної регресії взаємозв'язок між даними моделюється за допомогою лінійних функцій, а невідомі параметри моделі оцінюються за вхідними даними. Подібно до інших методів регресійного аналізу лінійна регресія повертає розподіл умовної імовірності y в залежності від X , а не розподіл спільної імовірності y та X , що стосується області мультиваріативного аналізу.

При розрахунках параметрів моделі лінійної регресії зазвичай застосовується метод найменших квадратів (МНК), але також можуть бути використані інші методи. Так само метод найменших квадратів може бути використаний і для нелінійних моделей. Тому МНК та лінійна регресія хоч і є тісно пов'язаними, але не є синонімами [6].

Регресійний аналіз використовується для прогнозу, аналізу часових рядів, тестування гіпотез та виявлення прихованих взаємозв'язків в даних.

Для нашої ж задачі ми будемо використовувати метод найменших квадратів, так, як його реалізація не є громіздкою, а результати в загалом видаватимуться адекватними в разі використання великої кількості даних.

Лінійна регресійна модель має наступний вигляд:

$$y = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k + u \quad (4.1.1),$$

де y — залежна змінна;

(x_1, x_2, \dots, x_n) — незалежні змінні;

u — випадкова похибка, розподіл якої в загальному випадку залежить від незалежних змінних, але математичне очікування якої рівне нулю.

Згідно з моделлю (4.1.1) математичне очікування залежної змінної є лінійною функцією незалежних змінних:

$$E(y) = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k + u \quad (4.1.2)$$

Вектор параметрів $(\beta_0, \beta_1, \dots, \beta_k)$ є невідомим і задача лінійної регресії полягає в оцінці цих параметрів на основі деяких експериментальних значень y і (x_1, x_2, \dots, x_n) . Тобто для деяких n експериментів є відомі значення $\{y_i, y_{i1}, \dots, y_{ip}\}_{i=1}^n$ незалежних змінних і відповідне їм значення залежної змінної.

Згідно з визначенням моделі для кожного експериментального випадку залежність між змінними визначається формулою:

$$y_i = \beta_0 + \beta_1 x_1 + \dots + \beta_k x_k + u_i, \quad (4.1.3)$$

або у матричних позначеннях.

На основі цих даних потрібно оцінити значення параметрів $(\beta_0, \beta_1, \dots, \beta_k)$, а також розподіл випадкової величини u . Зважаючи на характеристики досліджуваних змінних, можуть додаватися різні додаткові специфікації моделі і застосовуватися різні методи оцінки параметрів. Серед найпоширеніших специфікацій лінійних моделей є класична модель лінійної регресії та узагальнена модель лінійної регресії.

В залежності від об'єктів, що досліджуються за допомогою лінійної регресії та конкретних цілей дослідження, можуть використовуватися різні методи оцінки невідомих коефіцієнтів. Так як у випадку розробки нашої системи ми обрали метод найменших квадратів ми маємо наступну формулу для оцінки невідомих коефіцієнтів [7]:

$$\hat{\beta} = \arg \min_{\beta} \sum_{i=1}^n \left| y_i - \beta_0 - \sum_{j=1}^k X_{ij} \beta_j \right|^2 = \arg \min_{\beta} \|y - X\beta\|^2 \quad (4.1.3)$$

В принципі, метод лінійної регресії використовується для багатьох прогнозів у сьогоденні. Тому варто його використати для прогнозування поведінки рейтингу гравця. Якщо правильно підготувати дані, то результати можуть бути дуже точними відносно інших моделей прогнозування.

4.2 Модель ARIMAX

Ця модель набрала багато популярності для рішення задач прогнозування часових рядів. Вона використовується у прогнозу на ринку, спорту, погоди та інших галузях нашого повсякденного життя.

Для кращого розуміння моделі слід розділити її на частини: Авторегресія(AR), інтеграція(I), змінне середнє(MA), розширення(X)

Модель авто регресії зводиться до тої самої моделі лінійної регресії, де нам потрібно деяким обраним методом знайти потрібні коефіцієнти та підставити їх у формулу залежності від усіх попередніх даних. В нашому випадку ми знову будемо використовувати метод найменших квадратів.

Модель інтеграції говорить, що нам значно простіше шукати не саме значення прогнозу, а лише його зміну. Звідси ми будемо отримувати новий часовий ряд, який являє собою різницю сусідніх значень [21].

Чому така операція називається інтеграцією? Насправді, наведена різниця є зворотною операцією, однак при використанні такого підходу для отримання в кінцевому підсумку майбутнього значення шуканого тимчасового ряду необхідно буде підсумовувати, тобто до поточного значення процесу додавати спрогнозовану різницю.

В моделі ж ARIMAX ми будемо використовувати інтеграцію другого порядку, тобто в такому випадку інтегрувати, тобто підсумувати, доведеться двічі, щоб спрогнозувати майбутнє значення шуканого тимчасового ряду.

Часто трапляється, що в досліджуваному процесі є значні відхилення. Як правило, вони досить складно досліджуються. Очевидно, що подібні викиди погано впливають на найближчі до них прогнози значення.

Для згладжування подібних піків і застосовується модель змінного середнього, яка, по суті справи, являє собою простий фільтр низьких частот. Змінне середнє для моменту t є алгебраїчним середнім кількох попередніх значень вихідного часового ряду $Z(t)$.

Таким чином, ясно, що ковзне середнє є фільтром, який дозволяє згладжувати викиди тимчасового ряду, які, в свою чергу, спрогнозувати досить складно або зовсім неможливо.

Модель розширення додає до моделі зовнішні фактори.

Ми використовуємо в моделі, як правило, майбутні значення зовнішніх факторів, які або спрогнозували заздалегідь, або отримали звідкись із зовні, наприклад, прогноз температури на завтра у Гісметео. Таким чином, кожен зовнішній фактор несе в собі помилку власного прогнозу і ця помилка може бути великою, а тому дуже негативно позначитися на точності прогнозування шуканого тимчасового ряду.

4.3 Модель нейронних мереж

Хоча загалом ми під нейронними мережами в основному ми розуміємо їх в задачах розпізнавання образів, або звуків, також слід зазначити їх можливу ефективність у випадку рішення задач прогнозування часових рядів.

Зазвичай при прогнозуванні часових рядів використовуються багатошарові, найчастіше тришарові, нейронні мережі прямого поширення. Виглядає така мережа в такий спосіб [15].

Потрібно розуміти, що входів і виходів мережі може бути різна кількість. Як правило, на вхід мережі подаються фактичні значення часового ряду, а також значення зовнішніх факторів, на виході виходить одне або кілька прогностичних значень процесу [7].

Загалом модель прийматиме наступний вигляд (рисунок 4.1):

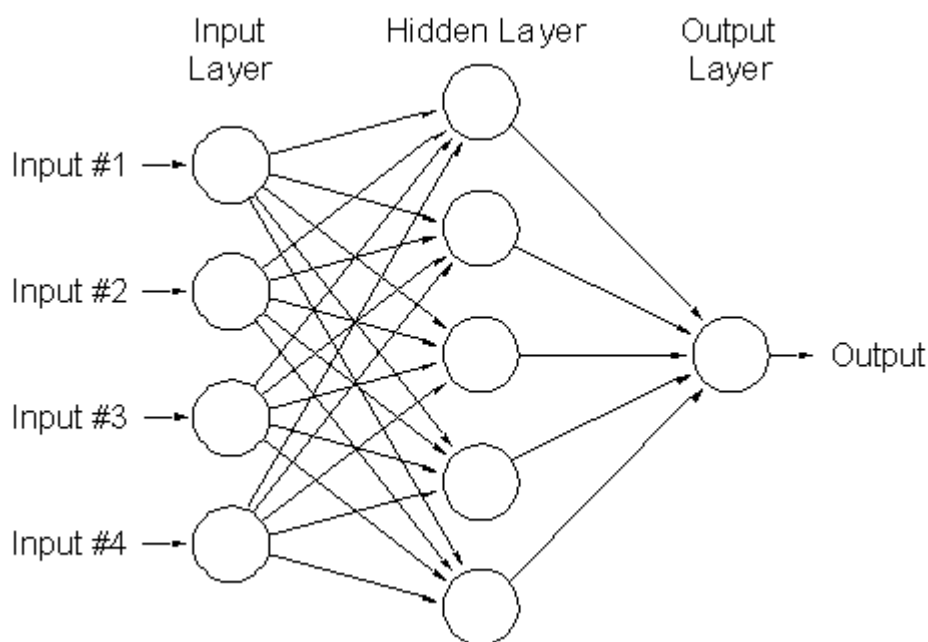


Рисунок 4.1 — Модель нейронних мереж

Таким чином, за допомогою нейронних мереж можливо моделювання нелінійної залежності майбутнього значення часового ряду від його фактичних значень і від значень зовнішніх факторів. Нелінійна залежність визначається структурою мережі і функцією активації [16].

Значним недоліком цієї моделі є неможливість розробнику отримати те, що відбувається всередині нейронних мереж. Ми формуємо входи, після цього розраховуємо виходи і просто зіставляємо одне з іншим. У нас немає можливості детально і покроково простежити те, як отримані на виході значення були розраховані. Цей режим виконання обчислень в «чорному ящику» надзвичайно ускладнює процес інтерпретації результатів і модифікації мережі - неясно, що в ній потрібно змінити, щоб стало точніше.

Та висока ефективність для рішення цього класу задач дозволяє ставити цю модель на одному рівні з моделлю ARIMAX.

4.4 Коефіцієнт детермінації

Окрім перевірки статистичної значущості оцінок параметрів парної лінійної регресії важливо також визначити якість моделі в цілому. Однією з найбільш ефективних оцінок адекватності регресійної моделі, мірою якості рівняння регресії, характеристикою її прогностичної сили є коефіцієнт детермінації. Цей коефіцієнт показує частину дисперсії пояснену рівнянням регресії:

$$R^2 = \frac{\hat{Var}(Y)}{Var(Y)} \quad (4.4.1)$$

Чим ближче R^2 до одиниці, тим краще регресія апроксимує емпіричні дані, тим тісніше точки спостереження примикають до лінії регресії. Якщо $R^2 = 1$, то емпіричні точки, що характеризують незалежну X та залежну Y змінну лежать на лінії регресії і між змінними існує лінійна функціональна залежність. Якщо $R^2 = 0$, то варіація залежної змінної повністю обумовлена дією неврахованих в моделі змінних, і лінія регресії паралельна осі абсцис [19].

Ця оцінка широко популярна серед аналітиків і тому дуже добре підходить для нашої задачі. Вона допоможе краще оцінити вказані раніше методи, збільшивши точність прогнозу.

4.5 Середнє абсолютне відхилення

Середнє абсолютне відхилення набору даних - це середня відстань між кожною точкою даних і середнім. Це дає нам уявлення про мінливість у наборі даних [18].

Формула обчислення даної оцінки має наступний вигляд:

$$MAD = median(|X_i - \tilde{X}|)$$

Середнє абсолютне відхилення є мірою статистичної дисперсії. Більш того, MAD є надійною статистикою, яка є більш стійкою до викидів у наборі даних, ніж стандартне відхилення. У стандартному відхиленні відстані від середнього квадрата квадратичні, тому великі відхилення важать більше, і, отже, викиди можуть сильно впливати на нього. У МАД відхилення невеликої кількості викидів не мають значення.

Оскільки MAD є більш надійною оцінкою масштабу, ніж дисперсія вибірки або стандартне відхилення, вона краще працює з розподілами без середнього значення або дисперсії, наприклад розподілу Коші [20].

Висновки до розділу 4

У даному розділі було описано три існуючі технології для побудови взаємодії. Звичайно таких технологій набагато більше. По кожній технології були наведені приклади використання та детальний опис процесу передачі й отримання даних.

5 РОБОТА КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ

В цьому розділі приведені системні вимоги для роботи з додатком та сценарії роботи користувача з ним.

5.1 Системні вимоги

Для підтримки мінімальної потужності роботи з системою персональний комп'ютер спеціаліста повинен мати процесор не гірше, ніж Intel ® Pentium ® / Celeron ® / Xeon™ або з тактовою частотою не менше 1,8 GHz або AMD 6 / Turion™ / Athlon™ / Duron™ / Sempron™ для користувачів процесорів від фірми AMD. Також комп'ютеру користувача повинно бути доступно не менше 2 Gb оперативної пам'яті та графічне ядро не гірше, ніж Intel ® HD Graphics 2000, що еквівалентно графічним картам з об'ємом пам'яті не менше, ніж 128 Mb.

На комп'ютері повинна бути встановлена система Windows 10, Windows 7, Windows XP Service Pack 2 або Windows Vista Service Pack 2. Також на жорсткому диску повинно бути не менше, ніж 300 Mb вільного місця для завантаження програмного продукту.

Користувачу також знадобиться встановлена база даних Postgre з назвою CsgoTest в якій повинні будуть зосереджені усі дані по гравцям. Усі дані беруться із цієї бази даних за допомогою запитів мовою PostgreSQL.

5.2 Робота аналітика з програмним продуктом

Інтерфейс додатку Windows Forms має наступний вигляд (рисунок 5.1):

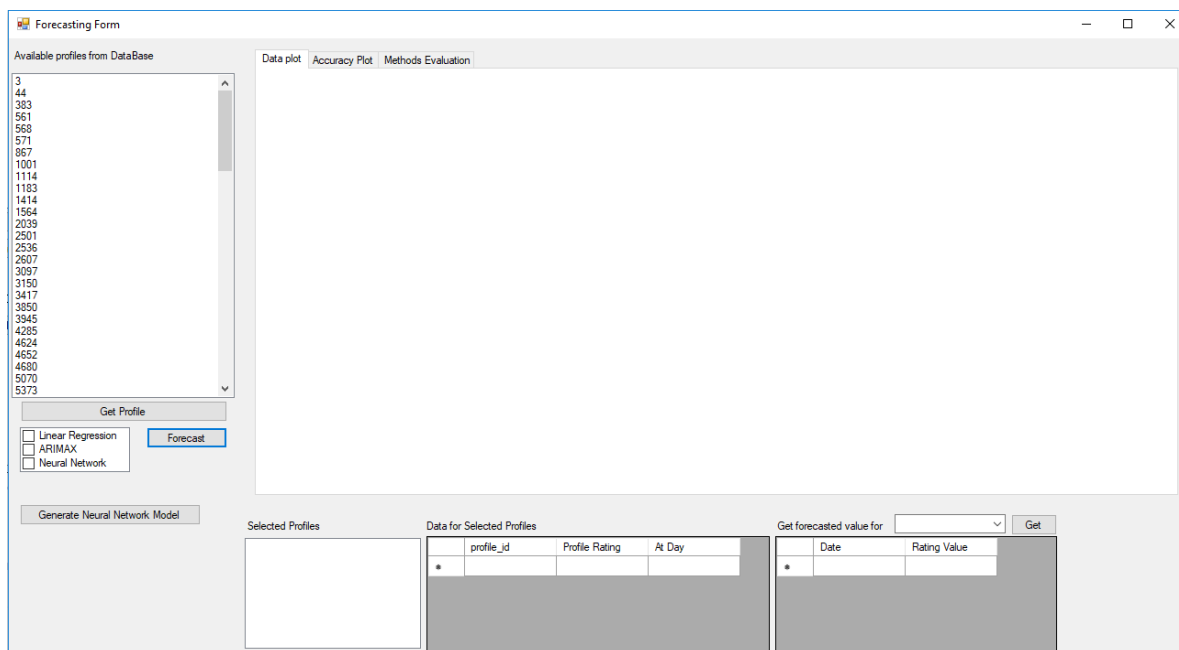


Рисунок 5.1 — Головне вікно додатку Windows Forms

Перед початком роботи спеціаліст-аналітик повинен підготувати базу даних з назвою CsgoTest, яка містить у собі таблицю profile_rating_log, з полями:

- profile_id;
- rating;
- date;

Вся ця інформація повинна бути отримана з баз даних сервісу DreamTeam та попередньо оброблена.

Для початку роботи користувачеві слід обрати необхідних гравців за їх профілями. Вони відтворені у вигляді індивідуальних кодів кожного профілю. Так Користувач системи може двійним потиском лівої кнопки миші по попередньо обраному коду на елементі listBox занести обраний профіль до прогнозованих. (рисунок 5.2).

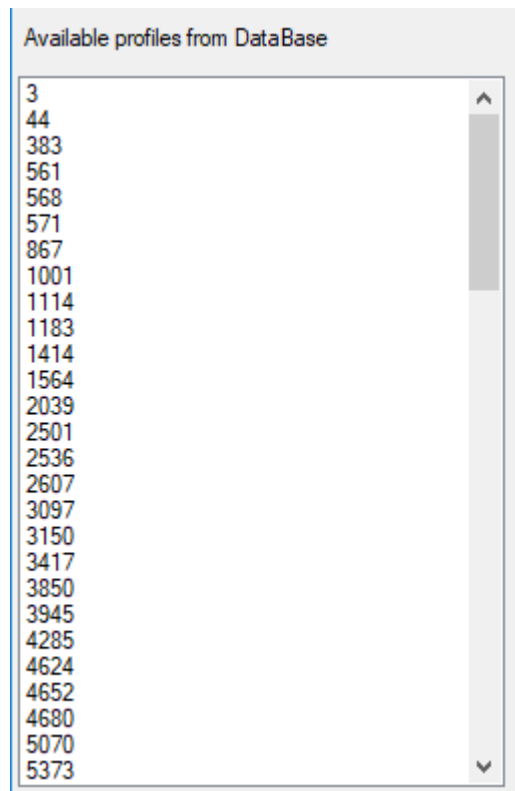


Рисунок 5.2 — Інструмент listBox з доступними профілями із бази даних

Після виконання певної вибірки гравців їх дані відображаються у правій частині вікна на графіку, та у елементах listBox і DataGridView, список усіх обраних профілів та таблиця з їх поточними даними (рисунок 5.3).

Для винесення певного профілю зі списку вибраних слід двійним потиском лівої кнопки миші по обраному коду у зазначеному елементу listBox.

Таблиця містить у собі усі дані, які були отримані з відповідної до цих профілів з бази даних полів таблиці profile_rating_log.

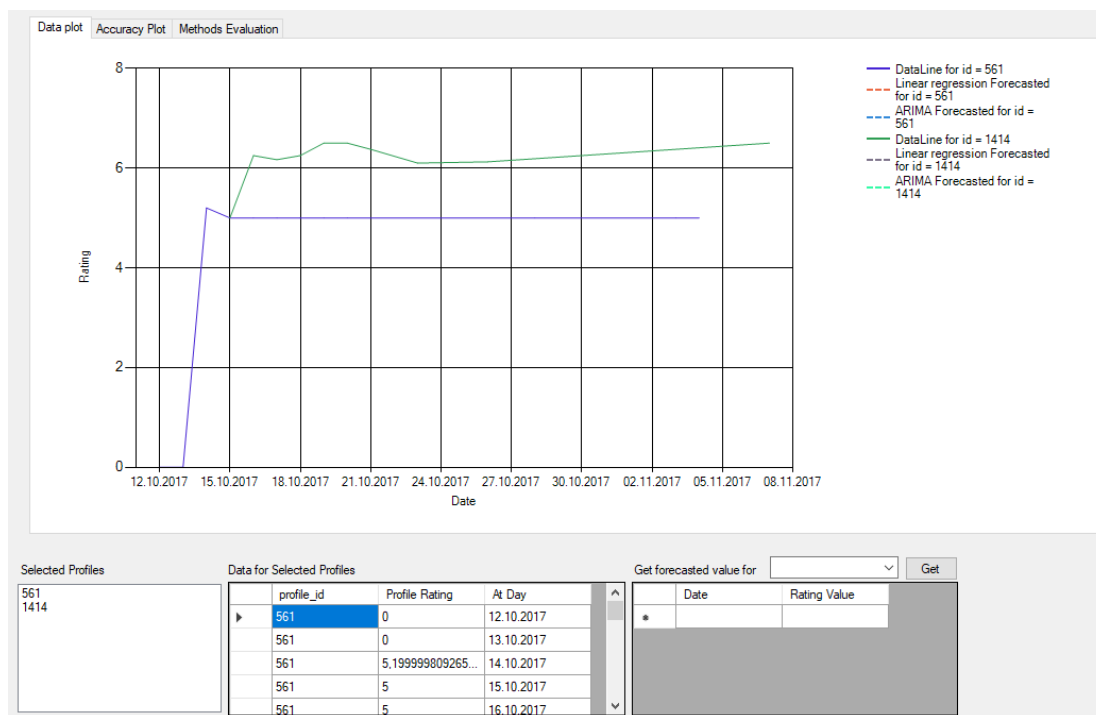


Рисунок 5.3 — Результат вибірки профілей гравців

Система виконує перевірку, щоб занесених до розглянутих профілів не було більше ніж можлива кількість у команді, яка становить 5 гравців (рисунок 5.4).

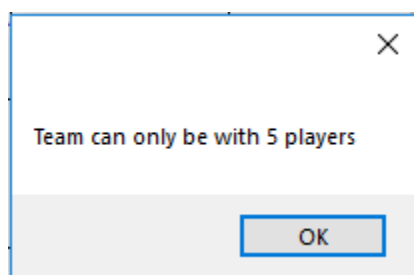


Рисунок 5.4 — Вікно помилки при спробі додати ще один профіль

Наступним кроком для аналітика буде вибірка потрібних йому моделей прогнозування та натискання кнопки “Forecast” для початку прогнозу (рисунок 5.5).

У результаті виконання прогнозу головний графік оновиться. До даних будуть також додані графіки прогнозу для кожного із методів (рисунок 5.6), будуть побудовані графік точності для методу лінійної регресії (рисунок 5.7) та змодельовані таблиці метрик для кожного з методів (рисунок 5.8).

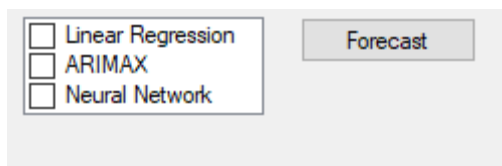


Рисунок 5.5 — Вибірка моделей прогнозу та кнопка початку прогнозу

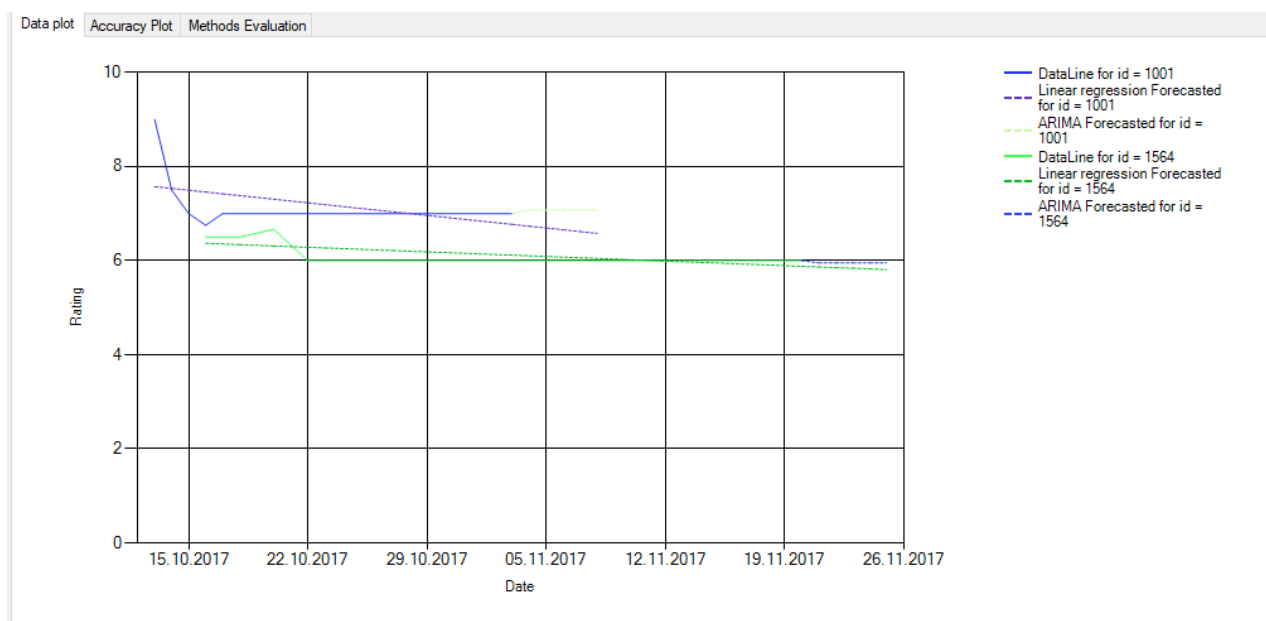


Рисунок 5.6 — Оновлений головний графік перегляду за даними

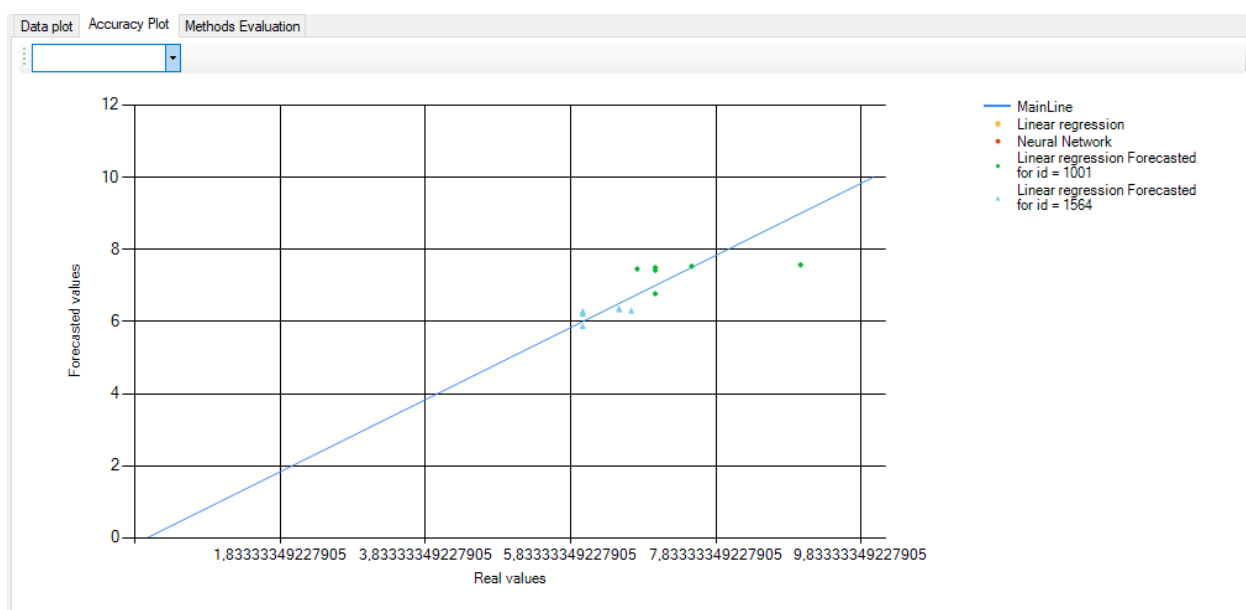


Рисунок 5.7 — Графік точності моделі лінійної регресії

Metric for Linear Regression and Neural Network				Metric for ARIMAX regression model			
	Name	R2	MADP		Name	ErrorVariance	LogLikelihood
►	Linear Regression for1001	0.130787624085211	0.444642410424319	►	ARIMA for1001	0.300786473138466	-4,2147566199983
	Linear Regression for1564	0.300526774704293	0.245489137399103		ARIMA for1564	0.0339443409205668	0,740017577251636
*				*			

Рисунок 5.8 — Таблиці метрик для обраних методів прогнозування

З цього користувач має змогу зробити оцінку окремо взятого гравця, або команди гравців, а також здійснити порівняння точності прогнозування кожного з обраних методів. Аналіз виконується по створеному графіку залежності рейтингу від дати та за допомогою метрик оцінювання моделей регресії.

Висновки до розділу 5

У даному розділі було описано три існуючі технології для побудови взаємодії. Звичайно таких технологій набагато більше. По кожній технології були наведені приклади використання та детальний опис процесу передачі й отримання даних.

ВИСНОВКИ

В ході виконання даної роботи було розроблено інформаційну систему для аналізу інформації щодо визначення та прогнозування статусу гравця\ команди гравців.

Додаток було написано на мові програмування C# з використанням графічного інтерфейсу Windows Forms.

Програму можна використовувати для моделювання поведінки рейтингу гравця, або вибірки із до п'ятьох гравців.

В ході роботи було проведено огляд та зроблено аналіз засобів вирішення проблеми задачі, що були використані для створення даного програмного забезпечення (середовища розробки Visual Studio 2017, інтерфейсу Windows Forms та відповідних бібліотек мови C#).

Відповідно до системи були зіставлені мінімальні технічні умови, для якісного її використання.

Система надає користувачу зручний інтерфейс для перегляду даних по гравцям та вивід усієї необхідної інформації по кожному із моделей прогнозування.

Результати роботи програмного продукту мають (20— 50% для мінімальної вхідної інформації, та 5— 15% для більш детальних вхідних даних). Ця програма робить можливим процес прогнозування розвитку рейтингу гравців.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Прогнозирование в спорте [Электронный ресурс] — Режим доступа: <https://mydocx.ru/11-48703.html>.
2. Анализ временных рядов [Электронный ресурс] — Режим доступа: <https://matlab.ru/solutions/application/computational-finance/time-series-analysis>.
3. Решение регрессионной задачи в среде Mathcad [Электронный ресурс] / В.Ф. Очков, Е.Г. Гридина, Л.А. Зайцева - НИУ «МЭИ» // с. 1 — 14 — Режим доступа: <http://tw.t.mpei.ac.ru/ochkov/Regress.pdf>.
4. Норман Дрейпер, Гарри Смит. Прикладной регрессионный анализ // 3-е издание. Перевод с англ. // Видання Вільямс, 2016.
5. ML.NET — An open source and cross-platform machine learning framework [Электронный ресурс] — Режим доступа: <https://dotnet.microsoft.com/apps/machinelearning-ai/ml-dotnet>.
6. Снитюк В.Е. ПРОГНОЗИРОВАНИЕ. Модели, Методы, Алгоритмы [Электронный ресурс] // Снитюк В.Е. // Учебный посібник. — Київ: Маклаут, 2008. — 367 с. — Режим доступа: https://www.researchgate.net/publication/281110176_PROGNOZIROVANIE_Modeli_Metody_Algoritmy.
7. Что влияет на точность прогнозирования временного ряда? [Электронный ресурс] Ирина Чучуева // Научная статья — Режим доступа: <http://www.mbureau.ru/blog/chto-vliyaet-na-tochnost-prognozirovaniya-vremennogo-ryada>.
8. Armstrong J.S. Forecasting for Marketing // Quantitative Methods in Marketing. London: International Thompson Business Press // 1999. — С. 92 — 119.
9. Jingfei Yang M. Sc. Power System Short-term Load Forecasting: Thesis for Ph.d degree. Germany, Darmstadt, Elektrotechnik und Informationstechnik der Technischen Universitat // 2006. — 139 с.

10. Microsoft Business Intelligence Tools for Excel Analysts (WILEY) by Michael Alexander, Jared Decker & Bernard Wehbe // Майкл Александер, Джаред Деккер, Бернارد Вехбе, 2014. — 384 с.
11. Ханк Д.Э., Уичерн Д.У., Райтс А.Дж. Бизнес-прогнозирование. // С. 14 — 28 — М.: Вільямс, 2003.
12. Vikram Dayal. An Introduction to R for Quantitative Economics: Graphing, Simulating and Computing. // Спрінгер, 2015. — 109 с.
13. Practical Time Series Forecasting with R: A Hands-On Guide by Galit Shmueli //Axelrod Schnall Publishers, 2015. — 208 с.
14. Python for Data Science For Dummies by John Paul Mueller, Luca Massaron // 2—ге видання. // 2019. — 324 с.
15. Gheyas I.A., Smith L.S. A Neural Network Approach to Time Series Forecasting // Proceedings of the World Congress on Engineering, London, [Електронний ресурс]. 2009, Видання 2 Р. 1292 — 1296. — Режим доступу: www.iaeng.org/publication/WCE2009/WCE2009_pp1292-1296.pdf
16. Morariu N., Iancu E., Vlad S. A neural network model for time series forecasting // Romanian Journal of Economic Forecasting. 2009, No. 4. P. 213 — 223.
17. Mazengia D.H. Forecasting Spot Electricity Market Prices Using Time Series Models: Thesis for the degree of Master of Science in Electric Power Engineering. Gothenburg, Chalmers University of Technology // 2008. — 89 с.
18. Mahfoud S., Mani G. Machine Learning Essentials: Practical Guide in R // Видання 10, 1996. — Р. 543 — 565.
19. Singh S. Metric Handbook: Planning and Design Data// Видання 3. 2008. — 856 с.
20. Чучуева И.А. Модель экстраполяции по максимуму подобия (ЭМП) для временных рядов цен и объемов на рынке на сутки вперед ОРЭМ (Оптовом рынке электроэнергии и мощности) // Наука и образование [Електронний ресурс]. № 1. 2010. — 105с. — Режим доступу: <http://technomag.edu.ru/doc/135870.html>

21. Чучуева И. А., Павлов Ю. Н. Сезонно-регрессионная модель прогнозирования в решении задачи прогнозирования цен РСВ (рынок на сутки вперед) // Энерго-Info. 2009. №4. С. 46 — 49.
22. Bunnoon P., Chalermyanont K., Limsakul C. A Computing Model of Artificial Intelligent Approaches to Mid-term Load Forecasting: a state-of-the-art-survey for the researcher // IACSIT International Journal of Engineering and Technology. 2010, No.1. P. 94 — 100.
23. Modeling and Forecasting Electricity Loads and Prices: A Statistical Approach // Rafał Weron, Wrocław University of Technology. 2006, John Wiley & Sons, Inc.. С. 81 — 83.
24. Староверов Б.А., Изотов В.А., Мормылев М.А. Повышение точности оперативных прогнозов потребления электроэнергии с помощью нейронных сетей за счет объединения процессов классификации и аппроксимации суточных профилей // Вестник ИГЭУ. 2007, Выдания 4. С. 91 — 93.
25. Chuchueva I. The time series extrapolation model based on maximum likeness set // Математическое моделирование социальной и экономической динамики: труды III Международной конференции. М., 2010. С. 281— 283.
26. Инструментальные средства и математические модели прогнозирования остаточного ресурса по фактически измеряемым параметрам оборудования электростанций // Выдания Ленард, 2015. — 160 с.

Додаток А

Алгоритмізації та розробки алгоритму аналізу інформації щодо визначення та прогнозування статусу гравця\команди гравців

Специфікація

УКР.НТУУ"КПІ імені Ігоря Сікорського"_ТЕФ_АПЕПС_ТР5279_19Б

Аркушів 2

Київ 2019

Позначення	Найменування	Примітки
Документація		
УКР.НТУУ"КПІ імені Ігоря Сікорського"_ТЕФ_АПЕПС_ТР5279_19Б	Записка.docx	Текстова частина дипломної роботи
Компоненти		
УКР.НТУУ"КПІ імені Ігоря Сікорського"_ТЕФ_АПЕПС_ТР5279_19Б 12-1	Program.7z	Основний файл архіву, в якому лежать exe файл та необхідні бібліотеки
УКР.НТУУ"КПІ імені Ігоря Сікорського"_ТЕФ_АПЕПС_ТР5279_19Б 12-2	dreamteam_dump.tar .lzma	Файл для відновлення бази даних на Postgre

Додаток Б

Алгоритмізації та розробки алгоритму аналізу інформації щодо визначення та прогнозування статусу гравця\команди гравців

Лістинг програми

УКР.НТУУ”КПІ імені Ігоря Сікорського”_ТЕФ_АПЕПС_ТР5279_19Б

Аркушів 8

Київ 2019

Текст програми моделі нейронних мереж

```

using System;
using System.Collections.Generic;
using System.Globalization;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Microsoft.ML;
using Microsoft.ML.Data;
using Microsoft.ML.Trainers;
using PredictWinForm.NeuralNetwork;
using PredictWinForm.Types;

namespace PredictWinForm.Predict
{
    public class NeuralNet : IPredict
    {
        static readonly string _modelPath = Path.Combine(Environment.CurrentDirectory, "Data", "Model.zip");
        MLContext mlContext;
        ITransformer model;

        public void Train(List<Rating_Series> RSdata, out double R2, out double meanSquared)
        {
            List<RatingData> rd = new List<RatingData>();
            foreach (var rs in RSdata)
            {
                float[] _dates = new float[4];
                float[] _HistoricalRate = new float[4];
                int j = 0;
                for (int i = rs.ratings.Count-5; i < rs.ratings.Count-1; i++, j++)
                {
                    _dates[j] = float.Parse((rs.ratings[i].at_day - rs.ratings[rs.ratings.Count - 5].at_day).TotalDays.ToString(),
CultureInfo.InvariantCulture.NumberFormat);
                    _dates[j] = (float)rs.ratings[i].rating;
                }
                float CurrRate = (float)rs.ratings.Last().rating;
                rd.Add(new RatingData
                {
                    dates = _dates,
                    HistoricalRate = _HistoricalRate,
                    CurrentRate = CurrRate
                });
            }

            // Create MLContext
            mlContext = new MLContext();

            // Load Data
            IDataView data = mlContext.Data.LoadFromEnumerable<RatingData>(rd);

            //Split data
            DataOperationsCatalog.TrainTestData dataSplit = mlContext.Data.TrainTestSplit(data, testFraction: 0.2);
            IDataView trainData = dataSplit.TrainSet;
            IDataView testData = dataSplit.TestSet;

            // Define Data Prep Estimator
            // 1. Concatenate Size and Historical into a single feature vector output to a new column called Features
            // 2. Normalize Features vector
            IEstimator<ITransformer> dataPrepEstimator = mlContext.Transforms.Concatenate("Features", "dates",
"HistoricalRate")

```

```

        .Append(mlContext.Transforms.NormalizeMinMax("Features"));

// Create data prep transformer
ITransformer dataPrepTransformer = dataPrepEstimator.Fit(trainData);

// Apply transforms to training data
IDataView transformedTrainingData = dataPrepTransformer.Transform(trainData);

var pipeline = mlContext.Transforms.Concatenate("Features", "dates", "HistoricalRate")
    .Append(mlContext.Transforms.NormalizeMinMax("Features"))
    .Append(mlContext.Regression.Trainers.Sdca());

// Define StochasticDualCoordinateAscent regression algorithm estimator
var sdcaEstimator = mlContext.Regression.Trainers.Sdca();

// Build machine learning model
var trainedModel = pipeline.Fit(transformedTrainingData);

// Measure trained model performance
// Apply data prep transformer to test data
IDataView transformedTestData = dataPrepTransformer.Transform(testData);

// Use trained model to make inferences on test data
IDataView testDataPredictions = trainedModel.Transform(transformedTestData);

// Extract model metrics and get RSquared
RegressionMetrics trainedModelMetrics = mlContext.Regression.Evaluate(testDataPredictions);
R2 = trainedModelMetrics.RSquared;
meanSquared = trainedModelMetrics.MeanSquaredError;
double[] mtofile = new double[2] { R2, meanSquared };
mlContext.Model.Save(trainedModel, data.Schema, "model.zip");
File.WriteAllLines("metrics.txt", mtofile.Select(d => d.ToString()));
//model = trainedModel;
// Save Data Prep transformer
mlContext.Model.Save(dataPrepTransformer, data.Schema, "data_preparation_pipeline.zip");
}

public float Calculate(Rating_Series rs)
{
    try
    {
        float[] _dates = new float[4];
        float[] _HistoricalRate = new float[4];
        int j = 0;
        for (int i = 0; i < rs.ratings.Count - 1; i++, j++)
        {
            _dates[j] = float.Parse((rs.ratings[i].at_day - rs.ratings[0].at_day).TotalDays.ToString(),
CultureInfo.InvariantCulture.NumberFormat);
            _dates[j] = (float)rs.ratings[i].rating;
        }
        float CurrRate = (float)rs.ratings.Last().rating;
        RatingData rd = new RatingData
        {
            dates = _dates,
            HistoricalRate = _HistoricalRate,
            CurrentRate = CurrRate
        };

// Create MLContext
mlContext = new MLContext();

// Define data preparation and trained model schemas
DataViewSchema modelSchema;

```

```

        // Load data preparation pipeline and trained model
        ITransformer trainedModel = mlContext.Model.Load("model.zip", out modelSchema);

        PredictionEngine<RatingData, RatingPrediction> predictionEngine =
mlContext.Model.CreatePredictionEngine<RatingData, RatingPrediction>(trainedModel);

        RatingPrediction prediction = predictionEngine.Predict(rd);

        return prediction.rating;
    }
    catch(Exception er)
    {
        MessageBox.Show(er.Message);
    }

    throw new Exception("Cannot calculate forecasted value");
}

public void Evaluate()
{

}

public double GetY(double X)
{
    double rez = 0;

    return rez;
}
}
namespace PredictWinForm.NeuralNetwork
{
    public class RatingData
    {

        [LoadColumn(1, 4)]
        [VectorType(4)]
        public float[] dates { get; set; }

        [LoadColumn(5, 8)]
        [VectorType(4)]
        public float[] HistoricalRate { get; set; }

        [LoadColumn(9)]
        [ColumnName("Label")]
        public float CurrentRate { get; set; }

    }
}
namespace PredictWinForm.NeuralNetwork
{
    public class RatingPrediction
    {
        [ColumnName("Score")]
        public float rating;
    }
}

```


Текст програми моделі ARIMAX

```
using System;
using System.Collections.Generic;
using Extreme.Mathematics;
using Extreme.Statistics;
using Extreme.Statistics.TimeSeriesAnalysis;

namespace PredictWinForm.Predict
{
    public class ARIMA : IPredict
    {
        public ArimaModel model;

        public void Calculate(Vector<double> values)
        {
            model = new ArimaModel(values, 0, 1, 1);
            model.EstimateMean = true;
            model.Fit();
        }

        public double GetY(double X)
        {
            double rez = 0;

            return rez;
        }
    }
}
```

Текст програми моделі лінійної регресії

```
using Accord.Math;
using PredictWinForm.Types;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace PredictWinForm.Predict
{
    /// <summary>
    /// Use FillTheMatrix2 before Getting Y values
    /// </summary>
    public class LinearReg:IPredict
    {
        private double C0, C1;

        /// <summary>
        /// Get Coeffs by filling the matrixes and calculating their
        /// </summary>
        /// <param name="points">Points of regression?</param>
        public void FillTheMatrix2(List<PointD> points)
        {
            double s1 = 0, s2 = 0, s3 = 0, s4 = 0;
            foreach (var p in points)
            {
                s1 += p.X;
                s2 += p.X * p.X;
                s3 += p.Y;
                s4 += p.Y * p.X;
            }
        }
    }
}
```

```

    }

    double[,] A =
    {
        {points.Count, s1 },
        {s1,s2 }
    };

    double[,] B =
    {
        {s3},
        {s4}
    };

    double[,] X = Matrix.Solve(A, B, leastSquares: true);

    C0 = X[0, 0];
    C1 = X[1, 0];
}

```

```

    public double GetY(double X)
    {
        return (C0 + C1 * X);
    }
}

```

Текст програми класів для роботи з БД

```

using Npgsql;
using PredictWinForm.Types;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace PredictWinForm
{
    public class DBAccess
    {
        public string conn_param = "Server=127.0.0.1;Port=5432;User
Id=postgres;Password=7777777;Database=CsgoTestDB;";

        NpgsqlConnection conn;

        public List<Rating_Series> rate_series;

        public DBAccess()
        {
            conn = new NpgsqlConnection(conn_param);
        }

        public void SelectAllWhereCountIsBiggerThan(int count)
        {
            List<Rating> rll = new List<Rating>();
            try
            {
                conn.Open();
                //NpgsqlCommand com = new NpgsqlCommand("select * from games.profiles_ratings_log where
                \"profile_id\"=6306 order by \"at_day\"", conn);

```

```

        NpgsqlCommand com = new NpgsqlCommand("select * from games.profiles_ratings_log order by
        \"profile_id\", \"at_day\"", conn);
        NpgsqlDataReader reader;
        reader = com.ExecuteReader();
        while (reader.Read())
        {
            try
            {
                int id = reader.GetInt32(0);
                double rate = reader.GetDouble(1);
                DateTime day = reader.GetDateTime(2);
                rll.Add(new Rating(id, rate, day));
            }
            catch (Exception er)
            {
                MessageBox.Show(er.Message);
            }
        }

        var rez = rll.GroupBy(c => c.profile_id).Where(grp => grp.Count() > count);
        rate_series = new List<Rating_Series>();

        foreach (var l in rez)
        {
            var ls = l.ToList();
            rate_series.Add(new Rating_Series
            {
                profile_id = l.Key,
                ratings = ls
            });
        }

    }
    catch (Exception er)
    {
        MessageBox.Show(er.Message);
    }
    finally
    {
        conn.Close();
    }
}

public ProfileInfo GetProfileInfo(int profile_id)
{
    ProfileInfo rez = new ProfileInfo();
    try
    {

        string Nick = GenerateName(6);
        string Country;
        int Age;
        List<string> Lang;
        string rank;
        List<string> regs;
        string About;

        conn.Open();
        NpgsqlCommand com = new NpgsqlCommand("select * from games.profiles_ratings_log where \"profile_id\"=6306
order by \"at_day\"", conn);

```

```

NpgsqlDataReader reader;
reader = com.ExecuteReader();
while (reader.Read())
{
    try
    {

    }
    catch (Exception er)
    {
        MessageBox.Show(er.Message);
    }
}

}
catch (Exception er)
{
    throw new Exception("Cannot get Profile Data");
}
finally
{
    conn.Close();
}
return rez;
}

```

```

public static string GenerateName(int len)
{
    Random r = new Random();
    string[] consonants = { "b", "c", "d", "f", "g", "h", "j", "k", "l", "m", "n", "p", "q", "r", "s", "sh", "zh", "t", "v", "w",
"x" };
    string[] vowels = { "a", "e", "i", "o", "u", "ae", "y" };
    string Name = "";
    Name += consonants[r.Next(consonants.Length)].ToUpper();
    Name += vowels[r.Next(vowels.Length)];
    int b = 2; //b tells how many times a new letter has been added. It's 2 right now because the first two letters are already
in the name.
    while (b < len)
    {
        Name += consonants[r.Next(consonants.Length)];
        b++;
        Name += vowels[r.Next(vowels.Length)];
        b++;
    }

    return Name;
}
}
}

```

Додаток В

Алгоритмізації та розробки алгоритму аналізу інформації щодо визначення та прогнозування статусу гравця\команди гравців

Опис програмного коду

УКР.НТУУ”КПІ імені Ігоря Сікорського”_ТЕФ_АПЕПС_ТР5279_19Б

Аркушів 9

Київ 2019

АНОТАЦІЯ

Даний додаток містить опис програмної системи, розробленої для виконання аналізу інформації щодо визначення та прогнозування статусу гравців. Створена програмна система показує декілька методів прогнозування та виконуює такі завдання:

- введення даних для прогнозу;
- генерування моделей;
- виконання прогнозу;
- виведення результатів прогнозу;
- розрахунок оцінки моделей прогнозування.

Програма отримує дані за допомогою запитів до бази даних PostgreSQL завдяки бібліотеці Npgsql.

При розробці програмної системи використовувалась мова C# у наступному середовищі програмування: Microsoft Visual Studio 2015, із використанням Windows Forms API.

ЗМІСТ

1. Загальні відомості	64
2. Функціональне призначення.....	65
3. Опис логічної структури	66
4. Використовувані технічні засоби.....	67
5. Виклик і завантаження	68
6. Вхідні і вихідні данні.....	69

ЗАГАЛЬНІ ВІДОМОСТІ

У цьому додатку міститься опис програмної системи для аналізу інформації щодо її прогнозування. У додатку Б міститься програмний код головних модулів розроблених програмних систем.

Додаток працює в операційній системі Windows 7, Windows 8 та Windows10 і потребує встановленого на ПК пакету програм Microsoft Visual Studio 2015 та бази даних Postgre.

При розробці програмного додатку використовувалась мова C# з використанням середовища Microsoft Visual Studio 2015.

ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ

Розроблений додаток виконує завдання аналізу інформації щодо її прогнозування, а саме виконує прогнозування інформації по рейтингу гравців трьома методами:

- модель лінійної регресії;
- модель ARIMAX;
- модель нейронних мереж;

Також розроблені додатки можуть використовуватися в якості учбових матеріалів при оцінці якості обраних технологій прогнозування.

Функціональні обмеження на використання компонентів полягають лише в діапазоні введення даних для виконання прогнозування та їх коректності.

ОПИС ЛОГІЧНОЇ СТРУКТУРИ

Для реалізації системи в мові C# було виконано створення форми Windows Forms .NET. Засобами Visual Studio було використано спеціальний тип проекту під назвою Windows Forms Application, результатом чого стало створення головної форми для користувача.

У проекті було виконано розбиття структури на типи, які використовуються у тій, чи іншій моделі прогнозування, на класи, які містять в собі реалізацію цих моделей, та класу доступу до бази даних.

При запуску систем спочатку з'являється головне вікно. На цій формі можна виконувати навігацію між вкладками та виконувати прогноз за допомогою зручного інтерфейсу кнопок.

ВИКОРИСТОВУВАНІ ТЕХНІЧНІ ЗАСОБИ

Для забезпечення повноцінної роботи та досягнення високої ефективності роботи створених додатків для демонстрації аналізу інформації щодо визначення та прогнозування статусу гравців було обрано Visual Studio 2015 яка показала себе надійною та гнучкою середою розробки програм.

Розроблені додатки працюють в операційних системах Windows 7, Windows 8 та Windows10 і потребує встановленого на ПК пакету програм Microsoft Visual Studio 2015 та бази даних Postgre.

ВИКЛИК І ЗАВАНТАЖЕННЯ

Перед запуском програми слід завантажити та імпортувати базу даних з файлу “dreamteam_dump.tar.lzma”.

Розроблена система не потребує інсталяції, достатньо запустити виконуваний файл “PredictWinForm.exe” з архіву.

Після запуску користувач отримає доступ до головної форми програми, звідки може виконувати необхідні обчислення.

ВХІДНІ І ВИХІДНІ ДАНІ

Вхідними даними для розроблених додатків є інформація яка зчитується з бази даних Postgre.

Вхідні дані зберігаються у програму та представлені у вигляді об'єктів класу рейтингу гравців.

Вихідними даними є зпрогнозовані рейтинги гравців, які відображаються у вигляді чисел в таблиці, або на графіку залежності рейтингу від дати його отримання.

Вихідними даними є оцінка моделей прогнозування: коефіцієнт детермінованості та середнє абсолютне відхилення.